



IBM Systems - iSeries

i5/OS Commands

Starting with DSPNWSD (Display Network Server  
Description)

*Version 5 Release 4*







IBM Systems - iSeries

i5/OS Commands

Starting with DSPNWSD (Display Network Server  
Description)

*Version 5 Release 4*

**Note**

Before using this information and the product it supports, be sure to read the information in "Notices," on page 361.

**Second Edition (February 2006)**

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

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

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

---

# Contents

<b>Display Network Server Desc (DSPNWS)</b>	<b>1</b>
Parameters	1
Network server description (NWS)	1
Output (OUTPUT)	1
Option (OPTION)	1
Examples	2
Error messages	2

<b>Display NWS Storage Space (DSPNWSSTG)</b>	<b>3</b>
Parameters	3
Network server storage space (NWSSTG)	3
Output (OUTPUT)	3
Examples	3
Error messages	4

<b>Display Network Server Users (DSPNWSUSR)</b>	<b>5</b>
Parameters	5
Server (SERVER)	5
Server type (SVRTYPE)	5
Output (OUTPUT)	6
Examples	6
Error messages	6

<b>Display NWS User Attributes (DSPNWSUSRA)</b>	<b>7</b>
Parameters	7
User profile (USRPRF)	7
Profile type (PRFTYPE)	7
Option (OPTION)	8
Output (OUTPUT)	8
Examples	8
Error messages	8

<b>Display Object Authority (DSPOBJAUT)</b>	<b>9</b>
Parameters	9
Object (OBJ)	10
Object type (OBJTYPE)	10
ASP device (ASPDEV)	10
Output (OUTPUT)	11
File to receive output (OUTFILE)	11
Output member options (OUTMBR)	12
Authority type (AUTTYPE)	12
Examples	12
Error messages	13

<b>Display Object Description (DSPOBJD)</b>	<b>15</b>
Parameters	15
Object (OBJ)	16
Object type (OBJTYPE)	18
Detail (DETAIL)	18

ASP device (ASPDEV)	19
Output (OUTPUT)	20
File to receive output (OUTFILE)	20
Output member options (OUTMBR)	20
Examples	21
Error messages	21

<b>Display OptiConnect Link Sts (DSPOPCLNK)</b>	<b>25</b>
Parameters	25
Output (OUTPUT)	25
Examples	25
Error messages	25

<b>Display Optical (DSPOPT)</b>	<b>27</b>
Parameters	27
Volume identifier (VOL)	28
Optical device (DEV)	28
Data type (DATA)	28
Output (OUTPUT)	29
Path (PATH)	29
User space (USRSPC)	30
Replace user space (REPLACE)	30
File to receive output (OUTFILE)	30
Output member options (OUTMBR)	31
Examples	31
Error messages	32

<b>Display Optical Locks (DSPOPTLCK)</b>	<b>35</b>
Parameters	35
Type (TYPE)	35
Volume identifier (VOL)	35
Path (PATH)	36
Output (OUTPUT)	36
Examples	36
Error messages	36

<b>Display Optical Server (DSPOPTSVR)</b>	<b>37</b>
Parameters	37
Type (TYPE)	37
Output (OUTPUT)	37
Examples	38
Error messages	38

<b>Display Override (DSPOVR)</b>	<b>39</b>
Parameters	39
Overridden file name (FILE)	39
Merge overrides (MRGOVR)	40
Call level (LVL)	40
Activation group (ACTGRP)	40
Output (OUTPUT)	41
Examples	41
Error messages	41

<b>Display PDG Profile (DSPPDGPRF)</b>	<b>43</b>
Parameters	43
User (USER)	43
Output (OUTPUT)	43
Examples	44
Error messages	44
<b>Display Physical File Member (DSPPFM)</b>	<b>45</b>
Parameters	45
File (FILE)	45
Member (MBR)	45
From record (FROMRCD)	46
Examples	46
Error messages	46
<b>Display Program (DSPPGM)</b>	<b>49</b>
Parameters	49
Program (PGM)	49
Output (OUTPUT)	50
Detail (DETAIL)	50
Examples	51
Error messages	51
<b>Display Program Adopt (DSPPGMADP)</b>	<b>53</b>
Parameters	53
User profile (USRPRF)	53
Object type (OBJTYPE)	54
Output (OUTPUT)	54
File to receive output (OUTFILE)	54
Output member options (OUTMBR)	55
Examples	55
Error messages	55
<b>Display Program References (DSPPGMREF)</b>	<b>57</b>
Parameters	58
Program (PGM)	59
Output (OUTPUT)	60
Object type (OBJTYPE)	60
File to receive output (OUTFILE)	60
Output member options (OUTMBR)	61
Examples	61
Error messages	61
<b>Display Program Variable (DSPPGMVAR)</b>	<b>63</b>
Parameters	63
Program variables (PGMVAR)	63
Output format (OUTFMT)	64
Output (OUTPUT)	64
Program (PGM)	65
Char output start position (START)	65
Characters to display (LEN)	65
Recursion level (RCRLVL)	65
Examples	66
Error messages	66
<b>Display Problems (DSPPRB)</b>	<b>67</b>

Parameters	67
Problem identifier (PRBID)	68
Status type (STATUS)	68
Severity (SEV)	69
Period (PERIOD)	69
Hardware (HARDWARE)	70
Resource name (RESOURCE)	71
Product (LICPGM)	71
Function (FUNCTION)	71
Program (PGM)	74
Message identifier (MSGID)	74
Origin (ORIGIN)	74
Service number (SRVID)	75
User assigned (ASUSER)	75
Group assigned (GROUP)	75
Problem type (PRBTYPE)	76
Problem category (PRBCGY)	76
Output (OUTPUT)	76
File to receive output (OUTFILE)	77
Output member options (OUTMBR)	77
Type of information (TYPE)	78
Examples	78
Error messages	79

<b>Display PSF Configuration (DSPPSF CFG)</b>	<b>81</b>
Parameters	81
PSF configuration (PSFCFG)	81
Output (OUTPUT)	81
Examples	82
Error messages	82

<b>Display Program Temporary Fix (DSPPTF)</b>	<b>83</b>
Parameters	83
Product (LICPGM)	83
PTF numbers to select (SELECT)	84
Release (RLS)	84
Cover letter only (COVERONLY)	85
Output (OUTPUT)	85
File to receive output (OUTFILE)	85
Output member options (OUTMBR)	86
Examples	86
Error messages	86

<b>Display PTF Cover Letter (DSPPTFCVR)</b>	<b>89</b>
Parameters	89
Product (LICPGM)	89
PTF numbers to select (SELECT)	89
Release (RLS)	90
Attributes (CVRATR)	90
Cover letter language (CVRLTRLNG)	91
Output (OUTPUT)	91
Examples	92
Error messages	92

<b>Display Power On/Off Schedule (DSPPWRSCD)</b>	<b>93</b>
--	-----------

Parameters . . . . .	93
Output (OUTPUT) . . . . .	93
Start date (STRDATE) . . . . .	93
Days (DAYS) . . . . .	93
Examples . . . . .	94
Error messages . . . . .	94

<b>Display Record Locks (DSPRCDLCK) . . . . .</b>	<b>95</b>
Parameters . . . . .	95
Physical file (FILE) . . . . .	95
Member (MBR) . . . . .	95
Record number (RCDNBR) . . . . .	96
Output (OUTPUT) . . . . .	96
Examples . . . . .	96
Error messages . . . . .	96

<b>Dsp Recovery for Access Paths (DSPRCYAP) . . . . .</b>	<b>99</b>
Parameters . . . . .	99
Output (OUTPUT) . . . . .	99
ASP device (ASP) . . . . .	100
ASP group (ASPGRP) . . . . .	100
Examples . . . . .	100
Error messages . . . . .	101

<b>Display RDB Directory Entries (DSPRDBDIRE) . . . . .</b>	<b>103</b>
Parameters . . . . .	103
Entry (RDB) . . . . .	103
Output (OUTPUT) . . . . .	104
File to receive output (OUTFILE) . . . . .	104
Output member options (OUTMBR) . . . . .	104
Examples . . . . .	105
Error messages . . . . .	105

<b>Display Remote Definition (DSPRMTDFN) . . . . .</b>	<b>107</b>
Parameters . . . . .	107
System name (SYSTEM) . . . . .	107
Output (OUTPUT) . . . . .	108
File to receive output (OUTFILE) . . . . .	108
Output member options (OUTMBR) . . . . .	108
Examples . . . . .	109
Error messages . . . . .	109

<b>Display S/36 Configuration (DSPS36) . . . . .</b>	<b>111</b>
Parameters . . . . .	111
Output (OUTPUT) . . . . .	111
Examples . . . . .	111
Error messages . . . . .	111

<b>Display Save File (DSPSAVF) . . . . .</b>	<b>113</b>
Parameters . . . . .	113
Save file (FILE) . . . . .	113
Output (OUTPUT) . . . . .	113
Examples . . . . .	114
Error messages . . . . .	114

<b>Display Subsystem Description (DSPSBSD) . . . . .</b>	<b>117</b>
Parameters . . . . .	117
Subsystem description (SBSD) . . . . .	117
Output (OUTPUT) . . . . .	118
Examples . . . . .	118
Error messages . . . . .	118

<b>Display Security Attributes (DSPSECA) . . . . .</b>	<b>119</b>
Parameters . . . . .	119
Output (OUTPUT) . . . . .	119
Examples . . . . .	119
Error messages . . . . .	120

<b>Display Security Auditing (DSPSECAUD) . . . . .</b>	<b>121</b>
Parameters . . . . .	121
Output (OUTPUT) . . . . .	121
Examples . . . . .	121
Error messages . . . . .	121

<b>Display Software Resources (DSPSFWRSC) . . . . .</b>	<b>123</b>
Parameters . . . . .	123
Output (OUTPUT) . . . . .	123
File to receive output (OUTFILE) . . . . .	124
Output member options (OUTMBR) . . . . .	124
Examples . . . . .	124
Error messages . . . . .	125

<b>Display Sphere of Control Sts (DSPSOCSTS) . . . . .</b>	<b>127</b>
Parameters . . . . .	127
Output (OUTPUT) . . . . .	127
Detail (DETAIL) . . . . .	127
Examples . . . . .	128
Error messages . . . . .	128

<b>Display Spooled File (DSPSPLF) . . . . .</b>	<b>129</b>
Parameters . . . . .	129
Spooled file (FILE) . . . . .	129
Job name (JOB) . . . . .	129
Spooled file number (SPLNBR) . . . . .	130
Job system name (JOBSYSNAME) . . . . .	130
Spooled file created (CRTDATE) . . . . .	131
Fold records (FOLD) . . . . .	131
Examples . . . . .	131
Error messages . . . . .	132

<b>Display Service Attributes (DSPSRVA) . . . . .</b>	<b>135</b>
Parameters . . . . .	135
Examples . . . . .	135
Error messages . . . . .	135

<b>Display Service Agent (DSPSRVAGT) . . . . .</b>	<b>137</b>
Parameters . . . . .	137
Type (TYPE) . . . . .	137

Device (DEVICE) . . . . .	138
Category (CATEGORY) . . . . .	138
Sense byte format (SENSEFMT) . . . . .	138
System or logical partition (SYSNAME) . . . . .	139
Output (OUTPUT). . . . .	139
IBM Service representative (SRVREP) . . . . .	139
IBM Service telephone number (SRVTELNBR) . . . . .	139
Recommendations (RECOMMEND) . . . . .	140
Error log identifier (ERRLOGID) . . . . .	140
Examples. . . . .	140
Error messages . . . . .	140

**Display Service Program (DSPSRVPGM) . . . . . 141**

Parameters . . . . .	141
Service program (SRVPGM) . . . . .	141
Output (OUTPUT). . . . .	142
Detail (DETAIL) . . . . .	142
Examples. . . . .	143
Error messages . . . . .	143

**Display Service Status (DSPSRVSTS) 145**

Parameters . . . . .	145
Job name (JOB). . . . .	145
Duplicate job option (DUPJOB OPT) . . . . .	146
Examples. . . . .	146
Error messages . . . . .	146

**Display Server Auth Entries (DSPSVRAUTE) . . . . . 147**

Parameters . . . . .	147
User profile (USRPRF) . . . . .	147
Output (OUTPUT). . . . .	147
Examples. . . . .	147
Error messages . . . . .	148

**Display System Status (DSPSYSSTS) 149**

Parameters . . . . .	149
Output (OUTPUT). . . . .	149
Reset status statistics (RESET) . . . . .	149
Assistance level (ASTLVL) . . . . .	149
Examples. . . . .	150
Error messages . . . . .	150

**Display System Value (DSPSYSVAL) 151**

Parameters . . . . .	151
System value (SYSVAL) . . . . .	152
Output (OUTPUT). . . . .	176
Examples. . . . .	176
Error messages . . . . .	176

**Display Tape (DSPTAP) . . . . . 177**

Parameters . . . . .	177
Device (DEV) . . . . .	177
Volume identifier (VOL). . . . .	178
File label (LABEL). . . . .	178
Sequence number (SEQNBR) . . . . .	178
Data type (DATA) . . . . .	179
Output (OUTPUT). . . . .	179

End of tape option (ENDOPT). . . . .	179
File to receive output (OUTFILE). . . . .	180
Output member options (OUTMBR). . . . .	180
Examples. . . . .	180
Error messages . . . . .	181

**Display Tape Category (DSPTAPCGY) 183**

Parameters . . . . .	183
Output (OUTPUT). . . . .	183
File to receive output (OUTFILE). . . . .	183
Output member options (OUTMBR). . . . .	184
Examples. . . . .	184
Error messages . . . . .	184

**Display Tape Cartridge (DSPTAPCTG) 185**

Parameters . . . . .	185
Library device (DEV). . . . .	185
Cartridge ID (CTG) . . . . .	185
Category (CGY) . . . . .	186
Output (OUTPUT). . . . .	187
File to receive output (OUTFILE). . . . .	187
Output member options (OUTMBR). . . . .	188
Examples. . . . .	188
Error messages . . . . .	188

**Display Tape Status (DSPTAPSTS) 191**

Parameters . . . . .	191
Library device (DEV). . . . .	191
Output (OUTPUT). . . . .	191
File to receive output (OUTFILE). . . . .	192
Output member options (OUTMBR). . . . .	192
Output member options (OUTFILFMT). . . . .	193
Examples. . . . .	193
Error messages . . . . .	193

**Display Trademarks (DSPTM) . . . . . 195**

Parameters . . . . .	195
Examples. . . . .	195
Error messages . . . . .	195

**Display Trace (DSPTRC). . . . . 197**

Parameters . . . . .	197
Output (OUTPUT). . . . .	197
Program (PGM) . . . . .	197
Examples. . . . .	198
Error messages . . . . .	198

**Display Trace Data (DSPTRCDTA) . . . 199**

Parameters . . . . .	199
Output (OUTPUT). . . . .	199
Clear (CLEAR) . . . . .	199
Examples. . . . .	200
Error messages . . . . .	200

**Display User-Defined FS (DSPUDFS) 201**

Parameters . . . . .	201
User-defined file system (UDFS) . . . . .	201
Output (OUTPUT). . . . .	201
Examples. . . . .	202



Error messages . . . . .	202
<b>Display Upgrade Preparation (DSPUPGPRP) . . . . .</b>	<b>203</b>
Parameters . . . . .	203
Type of information (TYPE) . . . . .	203
Output (OUTPUT). . . . .	204
Examples. . . . .	204
Error messages . . . . .	204
<b>Display User Permission (DSPUSRPMN) . . . . .</b>	<b>205</b>
Parameters . . . . .	205
User profile (USER) . . . . .	205
Granted to/from (GRANTED). . . . .	205
Output (OUTPUT). . . . .	205
Examples. . . . .	206
Error messages . . . . .	206
<b>Display User Profile (DSPUSRPRF) . . . . .</b>	<b>207</b>
Parameters . . . . .	207
User profile (USRPRF) . . . . .	208
Type of information (TYPE) . . . . .	208
Output (OUTPUT). . . . .	209
File to receive output (OUTFILE). . . . .	209
Output member options (OUTMBR). . . . .	210
Examples. . . . .	210
Error messages . . . . .	210
<b>Display User Print Info (DSPUSRPTI) . . . . .</b>	<b>213</b>
Parameters . . . . .	213
User (USER). . . . .	213
Output (OUTPUT). . . . .	213
Examples. . . . .	213
Error messages . . . . .	214
<b>Display Work Station User (DSPWSUSR) . . . . .</b>	<b>215</b>
Parameters . . . . .	215
Output (OUTPUT). . . . .	215
Examples. . . . .	215
Error messages . . . . .	215
<b>Duplicate Optical (DUPOPT) . . . . .</b>	<b>217</b>
Parameters . . . . .	217
From volume identifier (FROMVOL) . . . . .	217
To volume identifier (TOVOL). . . . .	218
New volume identifier (NEWVOL) . . . . .	218
Clear (CLEAR) . . . . .	218
From device (FROMDEV) . . . . .	219
To device (TODEV) . . . . .	219
Allow media errors (ALWMEDERR). . . . .	219
Output (OUTPUT). . . . .	219
From end of media option (FROMENDOPT) . . . . .	220
To end of media option (TOENDOPT) . . . . .	220
Examples. . . . .	220
Error messages . . . . .	221
<b>Duplicate Tape (DUPTAP) . . . . .</b>	<b>225</b>

Parameters . . . . .	225
From device (FROMDEV) . . . . .	226
To device (TODEV) . . . . .	226
From sequence number (FROMSEQNBR) . . . . .	226
To sequence number (TOSEQNBR) . . . . .	227
From volume identifier (FROMVOL) . . . . .	227
To volume identifier (TOVOL). . . . .	227
Tape density (TODENSITY) . . . . .	228
Data compaction (COMPACT). . . . .	232
Files to duplicate (FILES) . . . . .	232
User label program (USRLBLPGM) . . . . .	233
From device end option (FROMENDOPT). . . . .	233
To device end option (TOENDOPT) . . . . .	234
Check for active files (CHECK) . . . . .	234
File expiration date (EXPDATE) . . . . .	234
Examples. . . . .	235
Error messages . . . . .	235
<b>Edit Authorization List (EDTAUTL) . . . . .</b>	<b>237</b>
Parameters . . . . .	237
Authorization list (AUTL) . . . . .	237
Examples. . . . .	237
Error messages . . . . .	237
<b>Edit Backup List (EDTBCKUPL) . . . . .</b>	<b>239</b>
Parameters . . . . .	239
Backup list (BCKUPL) . . . . .	239
Examples. . . . .	239
Error messages . . . . .	239
<b>Edit CHPND Constraints (EDTGPCST) . . . . .</b>	<b>241</b>
Parameters . . . . .	241
Examples. . . . .	241
Error messages . . . . .	241
<b>Edit DL File Attributes (EDTDLFA) . . . . .</b>	<b>243</b>
Parameters . . . . .	243
Examples. . . . .	243
Error messages . . . . .	243
<b>Edit DLO Authority (EDTDLOAUT) . . . . .</b>	<b>245</b>
Parameters . . . . .	245
Document library object (DLO) . . . . .	245
Folder (FLR). . . . .	246
System object name (SYSOBJNAM) . . . . .	246
Examples. . . . .	246
Error messages . . . . .	246
<b>Edit Document (EDTDOC) . . . . .</b>	<b>249</b>
Parameters . . . . .	249
Document (DOC) . . . . .	249
Folder (FLR). . . . .	249
Display exit panel (EXITPNL) . . . . .	249
Examples. . . . .	250
Error messages . . . . .	250
<b>Edit File (EDTF) . . . . .</b>	<b>253</b>
Parameters . . . . .	253

Stream file, or (STMF) . . . . .	253
Data base file (FILE) . . . . .	253
File member (MBR) . . . . .	254
Examples . . . . .	254
Error messages . . . . .	254

<b>Edit DBCS Conversion Dict (EDTIGCDCT) . . . . .</b>	<b>257</b>
Parameters . . . . .	257
DBCS conversion dictionary (IGCDCT) . . . . .	257
Dictionary entry (ENTRY) . . . . .	257
Examples . . . . .	258
Error messages . . . . .	258

<b>Edit Library List (EDTLIBL) . . . . .</b>	<b>261</b>
Parameters . . . . .	261
Examples . . . . .	261
Error messages . . . . .	261

<b>Edit Object Authority (EDTOBJAUT) . . . . .</b>	<b>263</b>
Parameters . . . . .	263
Object (OBJ) . . . . .	264
Object type (OBJTYPE) . . . . .	264
ASP device (ASPDEV) . . . . .	264
Examples . . . . .	264
Error messages . . . . .	265

<b>Edit Questions and Answers (EDTQST) . . . . .</b>	<b>267</b>
Parameters . . . . .	267
Q/A database (QSTDB) . . . . .	267
Lib containing Q/A database (LIB) . . . . .	267
Examples . . . . .	268
Error messages . . . . .	268

<b>Edit Rebuild of Access Paths (EDTRBDAP) . . . . .</b>	<b>269</b>
Parameters . . . . .	269
Examples . . . . .	269
Error messages . . . . .	269

<b>Edit Recovery for Access Paths (EDTRCYAP) . . . . .</b>	<b>271</b>
Parameters . . . . .	271
Examples . . . . .	271
Error messages . . . . .	272

<b>Edit S/36 Program Attributes (EDTS36PGMA) . . . . .</b>	<b>273</b>
Parameters . . . . .	273
S/36 program (PGM) . . . . .	273
Examples . . . . .	273
Error messages . . . . .	274

<b>Edit S/36 Procedure Attributes (EDTS36PRCA) . . . . .</b>	<b>275</b>
Parameters . . . . .	275
S/36 procedure member (MBR) . . . . .	275

Source file (FILE) . . . . .	275
Examples . . . . .	276
Error messages . . . . .	276

<b>Edit S/36 Source Attributes (EDTS36SRCA) . . . . .</b>	<b>277</b>
Parameters . . . . .	277
S/36 source member (MBR) . . . . .	277
Source file (FILE) . . . . .	277
Examples . . . . .	278
Error messages . . . . .	278

<b>Edit Workstation Object Aut (EDTWSOAUT) . . . . .</b>	<b>279</b>
Parameters . . . . .	279
Workstation object type (WSOTYPE) . . . . .	279
Examples . . . . .	281
Error messages . . . . .	281

<b>Eject Emulation Output (EJTEMLOUT) . . . . .</b>	<b>283</b>
Parameters . . . . .	283
Emulation device (EMLDEV) . . . . .	284
Emulation location (EMLLOC) . . . . .	284
Print device (PRTDEV) . . . . .	284
Examples . . . . .	284
Error messages . . . . .	285

<b>Element Definition (ELEM) . . . . .</b>	<b>287</b>
Parameters . . . . .	287
Type of value (TYPE) . . . . .	288
Value length (LEN) . . . . .	290
Constant value (CONSTANT) . . . . .	291
Restricted values (RSTD) . . . . .	291
Default value (DFT) . . . . .	292
Valid values (VALUES) . . . . .	293
Relational expression (REL) . . . . .	293
Range of values (RANGE) . . . . .	294
Special values (SPCVL) . . . . .	294
Single values (SNGVAL) . . . . .	294
Minimum values required (MIN) . . . . .	295
Maximum values allowed (MAX) . . . . .	295
Allow unprintable characters (ALWUNPRT) . . . . .	296
Allow variable names (ALWVAR) . . . . .	296
Is ELEM a program (PGM) . . . . .	296
Is ELEM a data area (DTAARA) . . . . .	297
If a file parameter, how used (FILE) . . . . .	297
Full field required (FULL) . . . . .	297
Value an expression (EXPR) . . . . .	298
Varying length (VARY) . . . . .	298
Pass attribute byte (PASSATR) . . . . .	298
Case of value (CASE) . . . . .	299
CCSID of value (CCSID) . . . . .	299
Display input (DSPINPUT) . . . . .	299
Choice text (CHOICE) . . . . .	300
Choice program (CHOICEPGM) . . . . .	300
Initial prompt length (INLPMTLEN) . . . . .	301
Prompt text or message ID (PROMPT) . . . . .	301
Examples . . . . .	302
Error messages . . . . .	302

<b>Else (ELSE)</b> . . . . .	<b>303</b>
Parameters . . . . .	303
Command (CMD) . . . . .	303
Examples . . . . .	304
Error messages . . . . .	304
<b>Emulate Printer Keys (EMLPRTKEY)</b> . . . . .	<b>307</b>
Parameters . . . . .	307
Emulation device (EMLDEV) . . . . .	307
Emulation location (EMLLOC) . . . . .	308
Print device (PRTDEV) . . . . .	308
Emulated printer key (PRTKEY) . . . . .	308
Examples . . . . .	308
Error messages . . . . .	308
<b>End Agent Services (ENDAGTSRV)</b> . . . . .	<b>309</b>
Parameters . . . . .	309
Preferences file directory (PREFDIR) . . . . .	309
Examples . . . . .	309
Error messages . . . . .	310
<b>End ASP Balance (ENDASPBAL)</b> . . . . .	<b>311</b>
Parameters . . . . .	311
ASP number (ASP) . . . . .	311
ASP device (ASPDEV) . . . . .	311
Examples . . . . .	312
Error messages . . . . .	312
<b>End Batch Job (ENDBCHJOB)</b> . . . . .	<b>313</b>
Parameters . . . . .	313
Examples . . . . .	313
Error messages . . . . .	313
<b>End CHT Server (ENDCHTSVR)</b> . . . . .	<b>315</b>
Parameters . . . . .	315
Server (SERVER) . . . . .	315
Node (NODE) . . . . .	316
Examples . . . . .	316
Error messages . . . . .	316
<b>End Cleanup (ENDCLNUP)</b> . . . . .	<b>317</b>
Parameters . . . . .	317
Examples . . . . .	317
Error messages . . . . .	317
<b>End Cluster Node (ENDCLUNOD)</b> . . . . .	<b>319</b>
Parameters . . . . .	319
Cluster (CLUSTER) . . . . .	320
Node identifier (NODE) . . . . .	320
Option (OPTION) . . . . .	320
Examples . . . . .	320
Error messages . . . . .	320
<b>End Communications Server (ENDCMNSVR)</b> . . . . .	<b>321</b>
Parameters . . . . .	321
How to end (OPTION) . . . . .	321
Controlled end delay time (DELAY) . . . . .	321

Examples . . . . .	322
Error messages . . . . .	322
<b>End Communications Trace (ENDCMNTRC)</b> . . . . .	<b>323</b>
Parameters . . . . .	323
Configuration object (CFGOBJ) . . . . .	323
Type (CFGTYPE) . . . . .	323
Examples . . . . .	324
Error messages . . . . .	324
<b>End Commitment Control (ENDCMTCTL)</b> . . . . .	<b>325</b>
Parameters . . . . .	325
Examples . . . . .	325
Error messages . . . . .	325
<b>End Copy Screen (ENDCPYSCN)</b> . . . . .	<b>327</b>
Parameters . . . . .	327
Source device (SRCDEV) . . . . .	327
Examples . . . . .	327
Error messages . . . . .	327
<b>End Cluster Resource Group (ENDCRG)</b> . . . . .	<b>329</b>
Parameters . . . . .	329
Cluster (CLUSTER) . . . . .	329
Cluster resource group (CRG) . . . . .	330
Exit program data (EXITPGMDTA) . . . . .	330
Examples . . . . .	330
Error messages . . . . .	330
<b>End Controller Recovery (ENDCTLRCY)</b> . . . . .	<b>331</b>
Parameters . . . . .	331
Controller (CTL) . . . . .	331
Examples . . . . .	331
Error messages . . . . .	331
<b>End Debug Mode (ENDDDBG)</b> . . . . .	<b>333</b>
Parameters . . . . .	333
Examples . . . . .	333
Error messages . . . . .	333
<b>End Debug Server (ENDDBGSVR)</b> . . . . .	<b>335</b>
Parameters . . . . .	335
Examples . . . . .	335
Error messages . . . . .	335
<b>End Database Monitor (ENDDBMON)</b> . . . . .	<b>337</b>
Parameters . . . . .	337
Job name (JOB) . . . . .	338
Monitor ID (MONID) . . . . .	338
Comment (COMMENT) . . . . .	338
Examples . . . . .	339
Error messages . . . . .	340
<b>End Device Recovery (ENDDEVRCY)</b> . . . . .	<b>341</b>

Parameters . . . . .	341
Device (DEV) . . . . .	341
Examples . . . . .	341
Error messages . . . . .	341

**End Directory Shadowing (ENDDIRSHD) . . . . . 343**

Parameters . . . . .	343
How to end (OPTION) . . . . .	343
Controlled end delay time (DELAY) . . . . .	343
Examples . . . . .	344
Error messages . . . . .	344

**End Do Group (ENDDO) . . . . . 345**

Parameters . . . . .	345
Examples . . . . .	345
Error messages . . . . .	345

**End Disk Reorganization (ENDDSKRGZ) . . . . . 347**

Parameters . . . . .	347
Auxiliary storage pool ID (ASP) . . . . .	347
ASP device (ASPDEV) . . . . .	347
Examples . . . . .	348
Error messages . . . . .	348

**End EPM Environments (ENDEPMENV) . . . . . 349**

Error messages for ENDEPMENV . . . . .	349
Parameters . . . . .	349
Environment Name (EPMENV) . . . . .	349
Environment Number (ENVNBR) . . . . .	350
Examples . . . . .	350

Error messages . . . . .	350
--------------------------	-----

**End Group Job (ENDGRPJOB) . . . . . 351**

Parameters . . . . .	351
Group job (GRPJOB) . . . . .	351
Group job to be resumed (RSMGRPJOB) . . . . .	351
Job log (LOG) . . . . .	351
Examples . . . . .	352
Error messages . . . . .	352

**End Host Server (ENDHOSTSVR) . . . . . 355**

Error messages for ENDHOSTSVR . . . . .	355
Parameters . . . . .	355
Server type (SERVER) . . . . .	355
End active connections (ENDACTCNN) . . . . .	356
Examples . . . . .	356
Error messages . . . . .	356

**End Input (ENDINP) . . . . . 357**

Parameters . . . . .	357
Examples . . . . .	357
Error messages . . . . .	357

**End IP over SNA Interface (ENDIPSIFC) . . . . . 359**

Parameters . . . . .	359
Internet address (INTNETADR) . . . . .	359
Examples . . . . .	359
Error messages . . . . .	359

**Appendix. Notices . . . . . 361**

Trademarks . . . . .	362
Terms and conditions . . . . .	363

---

## Display Network Server Desc (DSPNWSD)

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

Parameters  
Examples  
Error messages

The Display Network Server Description (DSPNWSD) command displays a network server description object.

Top

---

### Parameters

Keyword	Description	Choices	Notes
NWSD	Network server description	<i>Name</i>	Required, Positional 1
OUTPUT	Output	*, *_PRINT	Optional, Positional 2
OPTION	Option	*ALL, *BASIC, *PORTS, *VRTETHPTH, *VRTETHSEC, *STGLNK, *STGPTH, *STGPTHQON, *STGPTHSEC, *TCPIP, *RSTDEV, *CLUINFO	Optional

Top

---

### Network server description (NWSD)

Specifies the name of the network server description to be displayed.

This is a required parameter.

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*     The output is displayed for interactive jobs or printed with the job's spooled output for  
-     non-interactive jobs.

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

Top

---

### Option (OPTION)

Specifies what information you want displayed. You can choose to display all information (\*ALL) or specific information.

\*ALL     All information concerning the network server is shown.

**\*BASIC**

Only basic characteristics of the network server are shown.

**\*PORTS**

Only the information concerning attached communication descriptions is shown.

**\*VRTETHPTH**

Only the information concerning the iSCSI virtual Ethernet paths of the network server are shown.

**\*VRTETHSEC**

Only the information concerning the iSCSI virtual Ethernet paths security of the network server are shown.

**\*STGLNK**

Only the information concerning linked client storage spaces is shown.

**\*STGPTH**

Only the information concerning the iSCSI storage paths of the network server are shown.

**\*STGPTHIQN**

Only the information concerning the iSCSI storage path qualified names of the network server are shown.

**\*STGPTHSEC**

Only the information concerning the iSCSI storage paths security of the network server are shown.

**\*TCPIP**

Only the information concerning TCP/IP configuration is shown.

**\*RSTDEV**

Only the restricted device resources associated with the network server description are shown.

**\*CLUINFO**

Only the information concerning cluster configuration is shown.

Top

---

## Examples

```
DSPNWSD NWS(D(SERVER1))
```

This command displays information about the network server description named SERVER1. Since no option was specified, all information is displayed. The information is shown at the work station display from which the command was entered. If the command was submitted from a batch job, the output from the command is printed with the job's spooled output.

Top

---

## Error messages

None

Top

---

## Display NWS Storage Space (DSPNWSSTG)

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

Parameters  
Examples  
Error messages

The Display Network Server Storage Space (DSPNWSSTG) command displays a network server storage space. Output is directed to a display or a spooled printer file as indicated on the OUTPUT parameter and by the job type. Information displayed includes the format of the storage space: its size, the amount of free space available, the auxiliary storage pool that it exists in, and the drive at which it is linked to an NWSD (network server description).

Top

---

### Parameters

Keyword	Description	Choices	Notes
NWSSTG	Network server storage space	<i>Name</i>	Required, Positional 1
OUTPUT	Output	*, *PRINT _	Optional

Top

---

### Network server storage space (NWSSTG)

Specifies the name of the storage space to be displayed.

Top

---

### Output (OUTPUT)

Specifies whether the output is shown on the display of the requesting work station or is printed with the job's spooled output.

\*     Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

**\*PRINT**

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

Top

---

### Examples

DSPNWSSTG   NWSSTG(STGSPACE1)

This command displays a network server storage space named STGSPACE1.

Top

---

## Error messages

None

[Top](#)



---

## Display Network Server Users (DSPNWSUSR)

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

Parameters  
Examples  
Error messages

The Display Network Server Users (DSPNWSUSR) command allows a user to retrieve a list of all of the logged on users of a particular server. This command can be used to display logged on users for an individual NetWare server.

Output is directed to a display or a spooled printer file as indicated by the OUTPUT parameter and job type.

Top

---

### Parameters

Keyword	Description	Choices	Notes
SERVER	Server	<i>Character value</i>	Required, Positional 1
SVRTYPE	Server type	<u>*NWSUSRA</u> , *NWSA, *NETWARE	Optional, Positional 2
OUTPUT	Output	*, *PRINT	Optional

Top

---

### Server (SERVER)

Specifies the name of a network server.

*server-name*

Specify the name of a specific, active server defined for the network.

Top

---

### Server type (SVRTYPE)

Specifies the server type to use when displaying users.

#### \*NWSUSRA

The server type specified in the network server user attributes (CHGNWSUSRA command) for the user profile running the DSPNWSUSR command is used to display the network users.

#### \*NWSA

The network server type specified in the network server attributes (CHGNWSA command) is used to display the network users.

#### \*NETWARE

Only those NetWare users associated with the specified network server are displayed.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting workstation or printed with the job's spooled output.

\*  
-       Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

### **\*PRINT**

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

Top

---

## Examples

```
DSPNWSUSR  SERVER(NWS1)  SVRTYPE(*NETWARE)
```

The above command will allow you to display all of the NetWare users currently active for the server named NWS1. If NWS1 does not exist or is not active, an error message ('server NWS1 not found' or 'server NWS1 not active') will be returned.

Top

---

## Error messages

### **\*ESCAPE Messages**

#### **CPFA43C**

Network server users not displayed.

Top

---

## Display NWS User Attributes (DSPNWSUSRA)

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

Parameters  
Examples  
Error messages

The Display Network Server User Attributes (DSPNWSUSRA) command displays the network server user attributes for a user profile.

Output is directed to a display or a spooled printer file as indicated by the OUTPUT parameter and job type.

Top

---

### Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	<i>Simple name</i> , <u>*CURRENT</u>	Optional, Positional 1
PRFTYPE	Profile type	<u>*USER</u> , *GROUP	Optional, Positional 2
OPTION	Option	<u>*ALL</u> , *WINDOWS, *NETWARE, *WINDOWSNT	Optional, Positional 3
OUTPUT	Output	<u>*</u> , *PRINT	Optional

Top

---

### User profile (USRPRF)

Specifies the name of a user or group profile.

#### \*CURRENT

The user profile attributes for the current user profile are displayed.

#### *user-name*

Specify the name of a user profile to be displayed.

Top

---

### Profile type (PRFTYPE)

Specifies whether the information to be displayed is for a user profile or for a group profile.

#### \*USER

The information to be displayed is for a user profile.

#### \*GROUP

The information to be displayed is for a group profile.

Top

---

## Option (OPTION)

Specifies the displays to be shown.

**\*ALL** All displays that apply to all server types are shown.

**\*WINDOWS or \*WINDOWSNT**

The displays that apply to the Windows servers are shown, but additional information displays are not shown.

**Note:** \*WINDOWS should be used in V5R4 and later releases. The \*WINDOWSNT value is supported for compatibility with releases prior to V5R4.

**\*NETWARE**

The displays that apply to the \*NETWARE server type are shown, but additional information displays are not shown.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting workstation or printed with the job's spooled output.

**\*** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

**\*PRINT**

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

Top

---

## Examples

```
DSPNWSUSRA  USRPRF(NWSUSR1)
```

If the job is interactive, this command displays the network server user attributes defined for user NWSUSR1 at the workstation. If the job is batch, the network server user attributes for user NWSUSR1 is printed with the job's spooled output.

Top

---

## Error messages

**\*ESCAPE Messages**

**CPFA453**

Network server user attributes not displayed.

Top

# Display Object Authority (DSPOBJAUT)

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

Parameters  
 Examples  
 Error messages

The Display Object Authority (DSPOBJAUT) command displays the list of authorized users of an object and their assigned authority. If the object is secured by an authorization list, the name of the authorization list is also displayed. The public authority and primary group authority are also shown.

If the user entering the command does not have object management (\*OBJMGT) authority to the object, only that user's name and authorities are shown. The names of the other users and their authorities for the object are not shown. If an object does not have an owner name associated with it, no authorities for the object are shown.

The following are shown for the specified object:

- The object name
- The name of the library containing the object
- The name of the object owner
- The object type
- A list of all the users who are authorized to use the object
- The authority that each user has for the object
- The authorization list name (if the object is secured by an authorization list)

**Restrictions:** You must have use (\*USE) authority to the auxiliary storage pool device if one is specified.

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, *M36, *M36CFG, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *NTBD, *NWID, *NWSCFG, *NWSD, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMQR, *QRYDFN, *RCT, *S36, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *TBL, *TIMZON, *USRIDX, *USRPRF, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 2
ASPDEV	ASP device	<i>Name, *_</i> , *SYSBAS	Optional

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *PRINT, *OUTFILE	Optional, Positional 3
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	
AUTTYPE	Authority type	*OBJECT, *FIELD, *ALL	Optional

Top

---

## Object (OBJ)

Specifies the object for which the authorized users and their authority are to be displayed.

This is a required parameter.

*name* Specify the name of the object.

### Qualifier 2: Library

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

### \*CURLIB

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

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

Top

---

## Object type (OBJTYPE)

Specifies the object type of the object whose authority is to be displayed, such as program (\*PGM), file (\*FILE), or library (\*LIB). To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt).

This is a required parameter.

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

---

## Output (OUTPUT)

Specifies where the output from the command is sent.

\*  
- The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

**\*PRINT**

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

**\*OUTFILE**

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

Top

---

## File to receive output (OUTFILE)

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

**Qualifier 1: File to receive output**

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

**Qualifier 2: Library**

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

**\*CURLIB**

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

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

**Note:** If a new file is created, the system uses QAOBJAUT in QSYS with a format name of QSYDSAUT as a model.

If AUTTYPE(\*FIELD) is specified for a \*FILE object and a new outfile is created, the system uses QAFLDAUT in QSYS with a format name of QSYDSFLD as a model.

Top

---

## Output member options (OUTMBR)

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

### Element 1: Member to receive output

#### \*FIRST

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

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

### Element 2: Replace or add records

#### \*REPLACE

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

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

Top

---

## Authority type (AUTTYPE)

Specifies whether object level authority, field level authority, or both object level and field level authority are displayed. Field level authority information only applies to \*FILE objects.

#### \*OBJECT

The object level authority information is displayed, placed in a spooled file, or placed in an outfile.

If OUTPUT(\*) is requested and the object is a file with field level authorities, the F16 key, Display Field Authorities, will be enabled on the display.

#### **\*FIELD**

The field level authority information is displayed, placed in a spooled file, or placed in an outfile.

This value is only valid if \*FILE is specified for the **Object type (OBJTYPE)** parameter.

**\*ALL** If OUTPUT(\*) is requested, the object level authority information is displayed. If there are field level authorities associated with the file, the F16 key, Display Field Authorities, will be enabled on the display. If OUTPUT(\*PRINT) is requested, the object level and field level authority data are included in the spooled file. AUTTYPE(\*ALL) is not valid with OUTPUT(\*OUTFILE).

This value is only valid if \*FILE is specified for the **Object type (OBJTYPE)** parameter.

Top

---

## Examples

### Example 1: Displaying Users and Authorities

```
DSPOBJAUT OBJ(ARLIB/PROG1) OBJTYPE(*PGM)
```

This command shows the authorized users and their authorities for the object named PROG1 to the user who entered the command, if that user has object management authority for the object. If the user does not have object management authority, only personal authorities are shown. PROG1 is a program (\*PGM) located in the library named ARLIB. The system assumes \* for the device that shows the output list. If



the command was entered in the batch subsystem, the output is placed in the default output queue for the job. If the command was entered in the interactive subsystem, the output is shown on the device where the user entered the command.

### Example 2: Printing List of Users

```
DSPOBJAUT OBJ(ARLIB/PROG2) OBJTYPE(*PGM) OUTPUT(*PRINT)
```

This command causes the list of authorized users of the program named PROG2 in the ARLIB library to be printed. If the user who enters the command does not have object management authority for the program, only that user's name and authorities are printed.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF2204

User profile &1 not found.

#### 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.

#### CPF2211

Not able to allocate object &1 in &3 type \*&2.

#### CPF2216

Not authorized to use library &1.

#### CPF224E

The AUTTYPE value of \*FIELD is not valid for file &1 in library &2.

#### CPF2283

Authorization list &1 does not exist.

#### CPF9843

Object &1 in library &3 type &2 cannot be accessed.

#### CPF9860

Error occurred during output file processing.

[Top](#)



---

## Display Object Description (DSPOBJD)

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

Parameters  
Examples  
Error messages

The Display Object Description (DSPOBJD) command shows the names and attributes of specified objects in the specified library or in the libraries of the library list for the current thread. The command can also show the names and attributes of libraries themselves.

Only the object attributes of each object are shown; the data attributes of data in the object, and the actual data in the object, are not shown. Also indicated is whether an object being shown has been damaged (possibly by a system failure) or is not able to be accessed because it is locked.

Any object for which you have some authority other than \*EXCLUDE authority can be shown by this command. Libraries for which you do not have execute (\*EXECUTE) authority cannot be shown, even if specified in the command. If only one object is to be shown, you can specify it by entering the object name, object type, the name of the library where it is located, and the auxiliary storage pool (ASP) device where the library is located. Depending on the specified library qualifier, either the first object found in the specified libraries is shown, or all objects in the specified libraries for which you have some authority are shown.

### Notes:

- For objects that are damaged or locked, the information shown, printed, or written to the output file is incomplete. If the description of the object is shown or printed, the text for the damaged or locked object indicates either the damaged or locked status. If the description of the object is written to an output file, an indication that the object is locked or damaged is included in the output file.
- The size of the library object shown does not include the sizes of the objects in the library. The total size of the library, including the sizes of the objects in the library, can be obtained using the Display Library (DSPLIB) command with OUTPUT(\*PRINT) or the Retrieve Library Description (QLIRLIBD) API.

### Restrictions:

1. You must have execute (\*EXECUTE) authority for the specified libraries. If you do not have \*EXECUTE authority for a library, none of its objects are shown.
2. You must have some authority other than \*EXCLUDE authority for each of the objects to be shown.
3. You must have either all object (\*ALLOBJ) or audit (\*AUDIT) special authority to see a value other than \*NOTAVL for the object auditing field.

Top

---

## Parameters

Keyword	Description	Choices	Notes
OBJ	Object	Single values: *ALLUSR, *IBM Other values: <i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *USRLIBL, *CURLIB, *ALL, *ALLUSR</i>	

Keyword	Description	Choices	Notes
OBJTYPE	Object type	Single values: *ALL Other values (up to 92 repetitions): *ALRTBL, *AUTL, *BNDDIR, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CNL, *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, *NWS, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMQR, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *TIMZON, *USRIDX, *USRPRE, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 2
DETAIL	Detail	* <b>BASIC</b> , *FULL, *SERVICE	Optional, Positional 3
ASPDEV	ASP device	Single values: *, *ALLAVL, *CURASPGRP, *SYSBAS Other values: <i>Element list</i>	Optional
	Element 1: Device	<i>Name</i>	
	Element 2: Search type	* <b>ASP</b> , *ASPGRP	
OUTPUT	Output	*, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , * <b>LIBL</b> , *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , * <b>FIRST</b>	
	Element 2: Replace or add records	* <b>REPLACE</b> , *ADD	

Top

## Object (OBJ)

Specifies which objects in the library or libraries are to have their object attributes shown. If library (qualifier 2) is not specified, \*LIBL is assumed, and all libraries in the library list for the current thread are searched for the objects. Objects in a library for which you do not have some authority are not shown. This parameter can be specified as a single value or as a list of one or two qualifiers.

This is a required parameter.

### Single values

#### \*ALLUSR

When the object name has a value of \*ALLUSR, the object type must be \*LIB. All user libraries in the auxiliary storage pools (ASPs) specified for the **ASP device (ASPDEV)** parameter are shown. Refer to \*ALLUSR in the description of library (qualifier 2) for a definition of user libraries.

**\*IBM** When the object name has a value of \*IBM, the object type must be \*LIB. All libraries in the

auxiliary storage pools (ASPs) defined by the **ASP device (ASPDEV)** parameter which are saved and restored using the SAVLIB and RSTLIB CL commands with \*IBM specified for the **Library (LIB)** parameter are shown.

### Qualifier 1: Object

- \*ALL** All objects in the libraries identified in library (qualifier 2) that are of the types specified for the **Object type (OBJTYPE)** parameter and for which you have some authority other than \*EXCLUDE authority are shown.
1. If the library (qualifier 2) is \*ALL, \*ALLUSR, or a library name, all objects of the specified type and that are in the specified libraries are shown.
  2. If the library (qualifier 2) is \*USRLIBL or \*LIBL and the object name is a specific name (not \*ALL and not a generic name), only one object type (and not \*ALL) can be specified for the OBJTYPE parameter and only the first object found is shown.

#### *generic-name*

Specify the generic name of the objects to be shown. A generic name is specified as a character string that contains one or more characters followed by an asterisk (\*). A generic name specifies all objects that have names with the same prefix as the generic object name for which you have some authority other than \*EXCLUDE authority.

*name* Specify the name of the object to be shown.

### Qualifier 2: Library

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

#### **\*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  QUSRIJS    QUSRVxRxMx
QGPL       QSRVAGT    QUSRINFSKR
QGPL38     QSYS2      QUSRNOTES
QMGTC      QSYS2xxxxx QUSROND
QMGTC2     QS36F      QUSRPOSGS
```

QMPGDATA	QUSER38	QUSRPOSSA
QMOMDATA	QUSRADSM	QUSRPYMSVR
QMOMPROC	QUSRBRM	QUSRDRARS
QPFRDATA	QUSRDIRCL	QUSRSYS
QRCL	QUSRDIRDB	QUSRVI

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

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

Top

---

## Object type (OBJTYPE)

Specifies which types of objects are shown. This parameter can be specified as a single value or as a list of one or more object types.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

This is a required parameter.

### Single values

**\*ALL** All types of objects with the specified object name are shown.

### Other values

#### *object-type*

Specify one or more values for the types of objects that are shown. All objects in the specified libraries, as well as libraries themselves, that have the object type(s) specified have their object attributes shown. If the library qualifier for the **Object (OBJ)** parameter is \*USRLIBL or \*LIBL and the object name qualifier is a specific name (not \*ALL and not a generic name), only one object type (and not \*ALL) can be specified here.

Top

---

## Detail (DETAIL)

Specifies which set of attributes is shown for each object.

**Note:** When the output is directed to a database file, this parameter is ignored and all object attribute information is written to the file.

### **\*BASIC**

The display or listing contains the name and a basic set of object attributes for each object.

**\*FULL** The display or listing contains the name and a full set of object attributes for each object (which includes the basic set of attributes).

### **\*SERVICE**

The display or listing contains the service-related attributes for each object.

---

## 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, \*LIBL, or \*USRLIBL, 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 (ASPs 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 (ASPs 2-32), and all available primary and secondary ASPs (ASPs 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 (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.

### 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 (ASPs 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.

---

## Output (OUTPUT)

Specifies where the output from the command is sent.

\*  
\_ The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

### \*PRINT

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

### \*OUTFILE

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

Top

---

## File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library.

**Note:** If a new file is created, the text describing that file is "Output file for DSPOBJD" and the public authority is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority. The database format (QLIDOBJD) of the output file is the same as that used in the IBM-supplied database file QADSPOBJ.

### Qualifier 1: File to receive output

**name** Specify the name of the file to which the output of the command 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 to find 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 to find the file.

Top

---

## Output member options (OUTMBR)

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

### Element 1: Member to receive output

#### \*FIRST

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

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



## Element 2: Replace or add records

### \*REPLACE

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

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

Top

---

## Examples

### Example 1: Displaying a Basic Description of Objects

```
DSPOBJD OBJ(X/PAY) OBJTYPE(*ALL)
```

A basic description of all the objects for which you have some authority (except exclude (\*EXCLUDE) authority) that are named PAY in library X are shown. Objects in the library for which you have no authority are not shown.

### Example 2: Displaying a Full Description of a Program

```
DSPOBJD OBJ(X/PAY) OBJTYPE(*PGM) DETAIL(*FULL)
```

A full description of the program named PAY in library X is shown. The display includes all the attributes of the program.

### Example 3: Displaying Program Information

```
DSPOBJD OBJ(*USRLIBL/PAY) OBJTYPE(*PGM)
```

This command shows information about the first program named PAY that is found in the user portion of the library list for the current thread.

### Example 4: Displaying a Basic Description of Files

```
DSPOBJD OBJ(Z/ABC*) OBJTYPE(*FILE)
```

A basic description of all of the files whose names begin with ABC (generic name) located in library Z for which you have some authority (except exclude (\*EXCLUDE) authority) are shown.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFB8ED

Device description &1 not correct for operation.

#### CPF2105

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

#### CPF2110

Library &1 not found.

#### CPF2113

Cannot allocate library &1.

#### CPF2114

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

**CPF2115**  
Object &1 in &2 type \*&3 damaged.

**CPF2121**  
One or more libraries cannot be accessed.

**CPF2123**  
No objects of specified name or type exist in library &2.

**CPF2124**  
No specified objects can be displayed from library &2.

**CPF2150**  
Object information function failed.

**CPF2176**  
Library &1 damaged.

**CPF2177**  
OBJTYPE value not compatible with OBJ value.

**CPF218C**  
&1 not a primary or secondary ASP.

**CPF218D**  
&1 not a primary ASP when \*ASPGRP specified.

**CPF2182**  
Not authorized to library &1.

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

**CPF326B**  
Damage to file &1 in library &2.

**CPF9809**  
Library &1 cannot be accessed.

**CPF9814**  
Device &1 not found.

**CPF9825**  
Not authorized to device &1.

**CPF9827**  
Object &1 cannot be created or moved into &2.

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

**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.

**CPF9899**

Error occurred during processing of command.

[Top](#)



---

## Display OptiConnect Link Sts (DSPOPCLNK)

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

Parameters  
Examples  
Error messages

The Display OptiConnect Link Status (DSPOPCLNK) command allows the user to view the status of either the fiber optic link or the high-speed link (HSL) connections between multiple systems. The system will determine if any high-speed links exist, and display them, otherwise the fiber optic links will be displayed.

[Top](#)

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *PRINT _	Optional, Positional 1

[Top](#)

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

Only Printing of the fiber optic links is available. If printed output of the link status is desired, this function is available through the hardware service manager screens.

\*        The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).  
\_

**\*PRINT**

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

[Top](#)

---

### Examples

```
DSPOPCLNK  OUTPUT(*PRINT)
```

This command will produce a listing showing the status of all connections that may be used by OptiConnect.

[Top](#)

---

### Error messages

None

[Top](#)



# Display Optical (DSPOPT)

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

Parameters  
 Examples  
 Error messages

The Display Optical (DSPOPT) command displays volume, directory, or file attributes depending on the value specified on the DATA parameter. The information can be printed, displayed, or written to an output file. If the output is directed to an outfile it will have one of the following record formats:

- When DATA(\*VOLATR) is specified the database file will have a record format named QAMODVA. The fields in the database are the same as the fields in the IBM-supplied format QAMODVA in file QAMODVAF located in library QSYS.
- When DATA(\*DIRATR) is specified the database file will have a record format named QAMODPA. The fields in the database are the same as the fields in the IBM-supplied format QAMODPA in file QAMODPAF located in library QSYS.
- When DATA(\*FILATR) is specified the database file will have a record format named QAMODFA. The fields in the database are the same as the fields in the IBM-supplied format QAMODFA in file QAMODFAF located in library QSYS.

**Restriction:** To use this command you must have \*USE authority to the authorization list securing the volumes being displayed.

Top

## Parameters

Keyword	Description	Choices	Notes
VOL	Volume identifier	Character value, *ALL, *MOUNTED	Required, Positional 1
DEV	Optical device	Name, *ALL	Optional, Positional 2
DATA	Data type	*VOLATR, *SAVRST, *FILATR, *DIRATR	Optional, Positional 3
OUTPUT	Output	*, *PRINT, *OUTFILE, *USRSPC	Optional
PATH	Path	Character value, *ALL	Optional
USRSPC	User space	Qualified object name	Optional
	Qualifier 1: User space	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
REPLACE	Replace user space	*YES, *NO	Optional
OUTFILE	File to receive output	Qualified object name	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
OUTMBR	Output member options	Element list	Optional
	Element 1: Member to receive output	Name, *FIRST	
	Element 2: Replace or add records	*REPLACE, *ADD	

---

## Volume identifier (VOL)

Specifies the identifier of the optical volume which contains the information being shown.

**\*ALL** The information is shown for all optical volumes on the specified directly-attached optical device (DEV parameter).

**\*MOUNTED**

The information is shown for the volume mounted on the specified device (DEV parameter).

**Note:** This value is valid only for directly-attached optical devices, not for library devices.

*volume-identifier*

Specify the identifier of an optical volume.

*generic\*-volume-identifier*

Specify the generic name of the volume identifier. 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 will result in the display of all volumes in directly attached optical devices which begin with the generic prefix and 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.

---

## Optical device (DEV)

Specifies the directly-attached optical device containing the optical volume whose information is displayed.

**Note:** This parameter is ignored when a specific volume name is specified on the VOL parameter.

**\*ALL** The volume attributes are displayed for optical volumes in all directly attached optical devices.

**Note:** This value is valid only when VOL(\*ALL) or VOL(generic\*) is specified.

*optical-device*

Specify the name of a directly-attached optical device.

---

## Data type (DATA)

Specifies the type of information that is displayed when specified on the PATH parameter. If DATA(\*SAVRST) is specified, the information includes a description of each object saved to the optical file and summary information about the saved objects. To determine whether the volume being displayed contains data in the basic stream file format or in the save and restore format, you can specify DATA(\*FILATR) and check the data file identifiers listed.

**\*VOLATR**

The volume attributes for the specified volume or volumes are displayed.

**\*DIRATR**

The directory attributes for the specified directory or directories are displayed.

**\*FILATR**

The file attributes for the specified file or files are displayed.



### \*SAVRST

The specified files contain save and restore data. Summary information is displayed for the command and each saved object.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station, printed with the job's spooled output, added to a database file or directed to a user space.

\*  
- Output requested by an interactive job is shown on the display. If the command is run as part of a batch job, the output is printed with the job's spooled output.

### \*PRINT

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

### \*OUTFILE

The output is directed to the database file specified on the OUTFILE parameter.

### \*USRSPC

The output is added to the user space specified on the USRSPC parameter.

Top

---

## Path (PATH)

Specifies the path name of the directory or the file on the volume being displayed. The DATA parameter indicates whether the directory or the file attributes are displayed. The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern can be specified in the last part of the path name. An asterisk (\*) matches any number of characters. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes.

**Note:** This parameter is ignored if DATA(\*VOLATR) is specified.

**\*ALL** \*ALL can have different meanings based on how parameter DATA is specified. Following is a list of combinations and their meanings.

- DATA(\*VOLATR) and PATH(\*ALL), the PATH parameter is ignored.
- DATA(\*SAVRST) and PATH(\*ALL), all save/restore files for all directories on the volume will be listed.
- DATA(\*SAVRST) and PATH(/directory1/\*ALL), all save/restore files in the path /directory1 will be listed.
- DATA(\*DIRATR) and PATH(\*ALL), all directories attributes for all directories on the volume will be listed.
- DATA(\*DIRATR) and PATH(/directory1/\*ALL), all directories attributes for directories in the path /directory1 will be listed.
- DATA(\*FILATR) and PATH(\*ALL), all file attributes for files in the root directory (/) of the volume will be listed.
- DATA(\*FILATR) and PATH(/directory1/\*ALL), all file attributes for files in /directory1 of the volume will be listed.

### *path-name*

Specify the fully qualified directory or file name which is to have its attributes listed.

### *generic\*-path-name*

Specify the generic name of the path identifier. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, /directory1/file\*. The asterisk

substitutes for any valid characters. A generic name will result in the display of all directories or files attributes which begin with the generic prefix and 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.

Top

---

## User space (USRSPC)

Specifies the user space to which the output of the display is added.

**Note:** This parameter is valid only if OUTPUT(\*USRSPC) is specified.

### Qualifier 1: User space

*user-space-name*

Specify the name of the user space that receives the output.

### 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.

*library-name*

Specify the name of the library to be searched.

Top

---

## Replace user space (REPLACE)

Indicates whether to replace an existing user space.

**Note:** This parameter is valid only if OUTPUT(\*USRSPC) is specified and is ignored if the user space is not found in the specified library.

**\*YES** The user space is replaced if found. The existing authorities of the original user space are retained, but the contents are replaced.

**\*NO** The user space is not replaced if found. The request ends and a message is sent to the job log indicating that the user space already exists in the library and cannot be created.

Top

---

## File to receive output (OUTFILE)

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

### Qualifier 1: File to receive output

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

### Qualifier 2: Library

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

**\*CURLIB**

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

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

Top

---

## Output member options (OUTMBR)

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

### Element 1: Member to receive output

**\*FIRST**

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

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

### Element 2: Replace or add records

**\*REPLACE**

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

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

Top

---

## Examples

### Example 1: Displaying Attributes for All Optical Volumes

```
DSPOPT VOL(*ALL) DEV(*ALL)
```

This command displays the volume attributes for all volumes in all local optical devices and libraries.

### Example 2: Displaying Attributes Using a Generic Search

```
DSPOPT VOL(PAY*) DATA(*DIRATR) PATH('/*')
```

This command displays the attributes for all directories in the root directory of all optical volumes beginning with the characters PAY.

### Example 3: Displaying Attributes of a Specific Directory

```
DSPOPT VOL(VOL01) DEV(OPT01) DATA(*FILATR)
      PATH('/DIR1/DIR2/*')
```

This command displays the file attributes for all files in the directory /DIR1/DIR2 of optical volume VOL01.

### Example 4: Displaying Save and Restore Data

DSPOPT VOL(\*MOUNTED) DEV(OPT01) DATA(\*SAVRST)  
PATH(' \*ALL')

This command displays the save and restore data for all files found on the optical volume mounted in device OPT01.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1247

Subsystem &1 cannot start prestart job &2.

#### CPF384C

Error occurred during CCSID conversion.

#### CPF3864

&2 &1 in &3 not restored.

#### CPF386A

File not found.

#### CPF5729

Not able to allocate object &1.

#### CPF9810

Library &1 not found.

#### CPF9815

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

#### CPF9845

Error occurred while opening file &1.

#### 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.

#### OPT1115

File not found.

#### OPT1125

File is in use.

#### OPT1135

Request failed because number of open files is at the limit.

#### OPT1140

Unexpected error occurred during file processing.

#### OPT1185

Cannot access held optical file.

#### OPT1205

Directory not found.

#### OPT1212

Directory in use.

- OPT1224**  
Path length exceeds the maximum of 256 bytes.
- OPT1247**  
User space &1 not created in library &2.
- OPT1255**  
File is corrupted.
- OPT1317**  
Directory name is too long.
- OPT1320**  
Optical volume &1 in use.
- OPT1325**  
Optical volume format not recognized.
- OPT1330**  
Optical volume not found or not useable.
- OPT1331**  
Optical volume &1 not found.
- OPT1340**  
Optical volume &1 not initialized.
- OPT1346**  
Operation not allowed to volume located in a remote optical device.
- OPT1360**  
Media directory corrupted on optical volume &1.
- OPT1427**  
A generic or \*ALL volume request is not allowed with DATA value.
- OPT1460**  
Optical volume &1 is not in an optical device.
- OPT1463**  
Operation not completed, optical volume is not a primary volume.
- OPT1489**  
Volume parameter is not permitted for device &1.
- OPT1530**  
&1 does not represent a valid optical device.
- OPT1555**  
Optical device &1 in use.
- OPT1605**  
Media or device error occurred.
- OPT1640**  
Error occurred reading files or directories.
- OPT1790**  
Operation not allowed or conflicts with another request.
- OPT1805**  
Error accessing optical volume index file.
- OPT1810**  
Error accessing optical directory index file.

- OPT1813**  
Unexpected error occurred.
- OPT1815**  
Internal program error occurred.
- OPT1820**  
Internal error occurred on optical device &1.
- OPT1821**  
Error occurred on optical device &1.
- OPT1825**  
Optical indexes are incorrect for optical device &1.
- OPT1860**  
Request to optical device &1 failed.
- OPT1861**  
No device description configured for resource &1.
- OPT1862**  
No active device description for resource &1.
- OPT1863**  
Optical libraries need to be reclaimed.
- OPT1872**  
Optical request timed out or was cancelled.
- OPT2040**  
Error accessing backup control file.
- OPT2301**  
Internal system object in use.
- OPT2420**  
Not authorized to optical volume &2.
- OPT2422**  
Not authorized to file or directory.
- OPT6713**  
Display volume details failed.
- OPT7740**  
User not authorized to object &2 in library &3 type &4.

Top

---

## Display Optical Locks (DSPOPTLCK)

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

Parameters  
Examples  
Error messages

The Display Optical Locks (DSPOPTLCK) command displays a list of locks held on an optical volume, directory, or file. If TYPE(\*JOB) is specified, this command displays a list of all jobs currently performing an optical request. The information can be printed or displayed.

This command does not identify any locks to volumes, directories, or files which are in remote optical servers. It also does not identify any jobs which are currently using a remote optical server. Use the Display Optical Server (DSPOPTSVR) command with TYPE(\*CONV) specified to determine if any jobs are currently using a remote optical server.

Top

---

### Parameters

Keyword	Description	Choices	Notes
TYPE	Type	*VOL, *DIR, *FILE, *JOB	Required, Positional 1
VOL	Volume identifier	<i>Character value</i>	Optional, Positional 2
PATH	Path	<i>Character value</i>	Optional
OUTPUT	Output	*, *_PRINT	Optional

Top

---

### Type (TYPE)

Specifies the type of locks to be displayed or printed.

- \*VOL Job information and locks on the specified optical volume are displayed or printed.
- \*DIR Job information and locks on the specified optical directory are displayed or printed.
- \*FILE Job information and locks on the specified optical file are displayed or printed.
- \*JOB Job information and locks on all jobs currently performing optical requests are displayed or printed.

Top

---

### Volume identifier (VOL)

Specifies the volume identifier of the optical volume for which the locks are listed.

Top

---

## Path (PATH)

Specifies the path name of the directory or file on the volume for which the locks are listed.

**Note:** This parameter is valid only if TYPE(\*DIR) or TYPE(\*FILE) is specified.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting workstation or printed with the job's spooled output.

\*     Output requested by an interactive job is shown on the display. If the command is run as part of  
-     a batch job, the output is printed with the job's spooled output.

**\*PRINT**

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

Top

---

## Examples

### Example 1: Displaying Locks on a File

```
DSPOPTLCK  TYPE(*FILE) VOLUME(VOL001)
           PATH('/PAYROLL/JAN1995')
```

This command displays the locks held on the file JAN1995 in the directory /PAYROLL on the VOL001 volume.

### Example 2: Displaying Locks for Active Jobs

```
DSPOPTLCK  TYPE(*JOB)
```

This command displays a list of active jobs performing optical requests.

Top

---

## Error messages

### \*ESCAPE Messages

#### OPT1318

File name is too long.

Top



---

# Display Optical Server (DSPOPTSVR)

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

Parameters  
Examples  
Error messages

The Display Optical Server (DSPOPTSVR) command displays information about the configuration of all optical servers added with the Add Optical Server (ADDOPTSVR) command. The information can be printed or displayed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
TYPE	Type	<u>*DEST</u> , *CONV	Optional, Positional 1
OUTPUT	Output	*, <u>_</u> , *PRINT	Optional, Positional 2

Top

---

## Type (TYPE)

Specifies the type of information to be displayed.

### \*DEST

The destination information is displayed. This information includes a listing of all of the optical servers accessible with the hierarchical file system APIs and the current status of each destination.

### \*CONV

The conversation information is displayed. This information includes a listing of all active optical conversations, the destination of each conversation, the jobs using the conversation, and the path of each open file.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting workstation or printed with the job's spooled output.

\*  
\_ The requested data is shown on the display station.

### \*PRINT

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

Top

---

## Examples

DSPOPTSVR TYPE(\*DEST)

This command displays the current status of each destination of all optical servers that have been started.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### **CPF4101**

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

#### **CPF6A1C**

Unable to add print function.

#### **CPF9845**

Error occurred while opening file &1.

#### **CPF9850**

Override of printer file &1 not allowed.

#### **CPF9851**

Overflow value for file &1 in &2 too small.

#### **OPT6710**

Optical server support is not active.

[Top](#)

---

## Display Override (DSPOVR)

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

Parameters  
Examples  
Error messages

The Display Override (DSPOVR) command displays file overrides at any active call level for a job. All file overrides, or file overrides for a specific file name, can be displayed.

The file overrides can be merged before being displayed. A merged file override is the result of combining all overrides for a file from call level one to the specified call level, producing the override which is applied when the file is opened at the specified call level.

A call level is associated with each call stack entry in the call stack. Calling a program or procedure adds another call stack entry to the call stack. When a program or procedure is called using the TFRCTL (Transfer Control) command, the call stack entry replaces a call stack entry that is already on the call stack; a new call level number is not created.

**Note:** This function can also be accessed through option 15 of the Work with Job (WRKJOB) command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FILE	Overridden file name	Name, <u>*ALL</u> , *PRTF	Optional, Positional 1
MARGOVR	Merge overrides	<u>*YES</u> , *NO	Optional
LVL	Call level	1-999, *, *JOB	Optional
ACTGRP	Activation group	Character value, *	Optional
OUTPUT	Output	*, *PRINT _	Optional

Top

---

### Overridden file name (FILE)

Specifies whether all file overrides, or file overrides for a specific file, are displayed.

**\*ALL** All the file overrides from call level one to the specified call level are displayed.

**\*PRTF** The \*PRTF file override, which exists in the call level where this command is entered, is displayed.

**name** Specify the name of the file for which all the file overrides, from call level one to the specified call level, are displayed.

Top

---

## Merge overrides (MRGOVR)

Specifies whether the file overrides are merged. Only those parameters on the overrides of the same type as the last override used for the merged override are used in determining the effective override for the specified call level.

**\*YES** The file overrides displayed are merged.

**\*NO** The file overrides displayed are not merged.

Top

---

## Call level (LVL)

Specifies the call levels of the file overrides displayed. There is a one-to-one correspondence between the call stack entries displayed on the call stack from the WRKJOB command and the call level for that call stack entry.

The first call stack entry name displayed on the call stack (at the top of the list) is the program or procedure at call level one. The second call stack entry name displayed is the program or procedure at call level two. The last call stack entry name displayed is the program or procedure at the highest call level for the job.

- If a merged file override is displayed, file overrides from call level one to the specified call level contribute to the creation of the merged file override.
- If **\*NO** is specified on the **Merge overrides (MRGOVR)** parameter and **\*ALL** is specified on the **File being overridden (FILE)** parameter, all file overrides (and the call levels at which they were found) from call level one to the specified call level are displayed.
- If **\*NO** is specified on the **Merge overrides (MRGOVR)** parameter, and a file override name is specified on the **File being overridden (FILE)** parameter, all file overrides for the file specified (and the call levels at which they were found) from call level one to the specified call level are displayed.

\*  
- The call level of the file override displayed is the call level of the program that called the DSPOVR command processing program. If this command is started by a call to QCMDEXC, the call level is the same call level as the caller of QCMDEXC. Overrides at call level numbers greater than 999 are not displayed.

**\*JOB** Only overrides at the job level with OVRSCOPE(\*JOB) specified are displayed.

**1-999** Specify the specific call levels of the file overrides to display. A specific call level is used to display file overrides at call levels lower than the call level at which the user is running.

Top

---

## Activation group (ACTGRP)

Specifies the level overrides to display for an activation group. When MRGOVR(\*YES) is specified, the activation group level overrides are processed after all call level overrides that are greater than or equal to the call level of the oldest procedure in the activation group are processed.

\*  
- The level overrides from the requester's activation group will be displayed.

*character-value*

Specify the name of the activation group that specifies activation group level overrides.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_ Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

### \*PRINT

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

Top

---

## Examples

### Example 1: Displaying Merged Overrides

```
DSPOVR FILE(REPORTS) MRGOVR(*YES) LVL(3) OUTPUT(*)
```

This command produces a display showing the merged override for the file REPORTS at call level 3 with text descriptions of each keyword and parameter. Applicable overrides at call levels 1, 2 and 3 are used to form the merged override.

### Example 2: Displaying File Overrides

```
DSPOVR FILE(REPORTS) MRGOVR(*NO) LVL(2) OUTPUT(*)
```

This command displays all file overrides for the file REPORTS up to call level 2. It produces a display showing the file name, the call level for which the override was requested, the type of override, and the override parameters. If no file overrides are found for the file up to and including the specified call level, escape message CPF9842 is sent.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF180C

Function &1 not allowed.

#### CPF1892

Function &1 not allowed.

#### CPF9842

Overrides not found for file &1.

#### 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.

**CPF9852**

Page size too narrow for file &1 in &2.

[Top](#)

---

## Display PDG Profile (DSPPDGPRF)

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

Parameters  
Examples  
Error messages

The Display Print Descriptor Group Profile (DSPPDGPRF) command displays the print descriptor group (PDG) and print descriptor name currently associated with the user.

### Restriction:

You must have \*OBJOPR authority to the user's profile.

Top

---

## Parameters

Keyword	Description	Choices	Notes
USER	User	Name, *CURRENT	Required, Positional 1
OUTPUT	Output	*, _ , *PRINT	Optional, Positional 2

Top

---

## User (USER)

Specifies the name of the user whose PDG profile is to be shown on the display.

The possible values are:

### \*CURRENT

Checks the user profile of the user of the current job.

### *user-name*

Specify the user whose PDG profile will be shown.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

The possible values are:

\*     The output is shown (if requested by an interactive job) or stored in a spooled file (if requested by a batch job).  
\_

### \*PRINT

The output is stored in a spooled file.

---

## Examples

DSPPDGPRF USER(TPDEXTER)

This command displays the print descriptor and the print descriptor group for user profile TPDEXTER.

Top

---

## Error messages

### \*ESCAPE Messages

**CPF2204**

User profile &1 not found.

**CPF2217**

Not authorized to user profile &1.

**CPF9820**

Not authorized to use library &1.

**CPF9871**

Error occurred while processing.

Top



---

## Display Physical File Member (DSPPFM)

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

Parameters  
Examples  
Error messages

The Display Physical File Member (DSPPFM) command displays a physical database file member. Records are shown in arrival sequence, even if the file has a keyed access path. You can page through the file, locate a particular record by record number, or specify a starting position in the record. You can also select a character or hexadecimal display of the records.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FILE	File	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: File	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MBR	Member	<i>Name, *FIRST, *LAST</i>	Optional, Positional 2
FROMRCD	From record	<i>Unsigned integer, 1, *END, *ALLDATA</i>	Optional

Top

---

### File (FILE)

Specifies the name and library of the physical file containing the member to be displayed.

This is a required parameter.

#### Qualifier 2: Library

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

#### \*CURLIB

The current library for the 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

---

### Member (MBR)

Specifies the name of the file member displayed.

#### \*FIRST

The first member of the specified physical file is displayed.

#### \*LAST

The last member of the specified physical file is displayed.

### *member-name*

Specify the name of the file member.

Top

---

## From record (FROMRCD)

Specifies which record in a physical file is shown on the top line of the initial display. If the specified record number is a deleted record, the display is positioned on the first record that follows the record that has been deleted. You can display as many records as needed using the page-up and page-down keys.

The possible values are:

- 1 Record number one, or the first non-deleted record, of the physical file is shown. If the file is a distributed file, this will be the first non-deleted record of the local member, and only local data will be shown.
- \*END** The last non-deleted record in the physical file is shown. The \*END value shows the last complete page so that the last record in the physical file appears at the bottom of the screen. If the file is a distributed file, this will be the last non-deleted record of the local member, and only local data will be shown.
- \*ALLDATA**  
All the data for a distributed file, including remote data, is shown. If \*ALLDATA is specified for a non-distributed file, it will be treated the same as FROMRCD(1).

### *record-number*

Specify the number of the record shown on the top line of the initial display. If the file is a distributed file, this will be the record number of the local member, and only local data will be shown.

Top

---

## Examples

### Example 1: Displaying the First File Member

```
DSPPFM FILE(TESTA)
```

This command shows the first member of a physical file named TESTA. The library list is used to locate the file.

### Example 2: Displaying a File Member

```
DSPPFM FILE(SAMPLE/TESTB) MBR(PROGRAM)
```

This command shows member PROGRAM of physical file TESTB in library SAMPLE.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF8056

File &1 in &2 not a physical file.

**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.

**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



---

## Display Program (DSPPGM)

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

Parameters  
Examples  
Error messages

The Display Program (DSPPGM) command displays information about a program. The display includes information about the compiler, the source from which the program was created, certain processing attributes of the program, the size of the program, and the number of parameters that must be passed to the program when it is called.

### Restrictions:

- You must have read (\*READ) authority to the program and execute (\*EXECUTE) authority to the library to use this command.
- You must have use (\*USE) authority to the program when DETAIL(\*MODULE) is specified.

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>	
OUTPUT	Output	<i>*, *PRINT</i>	Optional, Positional 2
DETAIL	Detail	Single values: <b>*ALL</b> Other values (up to 7 repetitions): *BASIC, *SIZE, *MODULE, *ACTGRPEXP, *ACTGRPIMP, *SRVPGM, *COPYRIGHT	Optional

Top

---

## Program (PGM)

Specifies the program for which information is displayed.

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.

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
- The output is shown (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

### \*PRINT

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

---

## Detail (DETAIL)

Specifies the type of information displayed for the service program. More than one value can be specified, but a list of values must not include \*ALL. \*ALL must be specified as a single value.

**Note:** DETAIL(\*ALL) or DETAIL(\*BASIC) are the only values valid for original program model (OPM) programs. All values other than \*ALL or \*BASIC are ignored for an OPM program.

**\*ALL** All of the DETAIL information types (\*BASIC, \*SIZE, \*MODULE, \*SRVPGM, \*ACTGRPEXP, \*ACTGRPIMP, and \*COPYRIGHT) are shown on the display. If the user has chosen the information to be displayed on the screen, the user can scroll through the information for each DETAIL, but will have to press Enter (or PF12) to go from DETAIL to DETAIL.

### \*BASIC

General program information is shown.

**\*SIZE** The size and size limits for this program are shown.

### \*MODULE

A list is shown of the module objects bound by this program. The library shown for each module is the library that the module was in when the program was first created. If the module has been replaced by a module from a different library, this library name remains the name of the library that the module was in when the program was created. To determine the source that the module was created from, use option 5=Display description to see the source file, library, and member names.

### \*SRVPGM

A list is shown of the service programs bound by this program.

### \*ACTGRPEXP

A list is shown of the data items exported to the activation group specified in the data export entry in the binding specifications.

### \*ACTGRPIMP

A list is shown of the imports that are resolved by weak exports that had been exported to the activation group directory.

### \*COPYRIGHT

A list is shown of the copyrights for this service program.

**Note:** The DETAIL values \*SIZE, \*MODULE, \*SRVPGM, and \*COPYRIGHT are valid only for integrated language environment (ILE) programs. Specifying one of these values for an original program model (OPM) program results in the \*BASIC information being shown.

---

## Examples

### Example 1: Displaying Program Information

```
DSPPGM PGM(LIB01/PAYROLL)
```

This command displays information about the program named PAYROLL in library LIB01. The display is shown at the display station if requested by an interactive job, or printed if requested by a batch job.

### Example 2: Printing Program Information

```
DSPPGM PGM(CUSINQ) OUTPUT(*PRINT)
```

This command displays information about a program named CUSINQ. The library list is used to find the program, and the information is printed.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2150

Object information function failed.

#### CPF2151

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

#### CPF8122

&8 damage on library &4.

#### CPF8123

Damage on object information for library &4.

#### CPF8129

Program &4 in &9 damaged.

#### CPF9803

Cannot allocate object &2 in library &3.

#### 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.

#### 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.

**CPF9830**

Cannot assign library &1.

**CPF9871**

Error occurred while processing.

[Top](#)



---

## Display Program Adopt (DSPPGMADP)

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

Parameters  
Examples  
Error messages

The Display Programs that Adopt (DSPPGMADP) command displays the objects that adopt the special and private authorities of the specified user profile. This is a convenient way to check security exposure due to program adoption.

### Restrictions:

- You must have object management (\*OBJMGT) authority to the user profile.
- The user profile specified on the command will be locked while the command is running. The lock prevents such things as objects having their ownership changed. If this profile owns a lot of objects, the profile could be locked for an extended period of time.

Top

---

## Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	<i>Name</i>	Required, Positional 1
OBJTYPE	Object type	Single values: <b>*ALL</b> Other values (up to 3 repetitions): *PGM, *SQLPKG, *SRVPGM	Optional, Positional 2
OUTPUT	Output	<b>*</b> , *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> , <b>*LIBL</b> , *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , <b>*FIRST</b>	
	Element 2: Replace or add records	<b>*REPLACE</b> , *ADD	

Top

---

## User profile (USRPRF)

Specifies the user profile whose authorities are adopted.

This is a required parameter.

**name** Specify the name of a user profile. Objects will be checked to see if they adopt special and private authorities from this user profile.

Top

---

## Object type (OBJTYPE)

Specifies the types of objects to be shown.

### Single values

**\*ALL** All objects that adopt the user profile specified on the **User profile (USRPRF)** parameter are shown.

### Other values (up to 3 repetitions)

**\*PGM** Only programs that adopt the specified user profile are shown.

#### **\*SQLPKG**

Only Structured Query Language (SQL) packages that adopt the specified user profile are shown.

#### **\*SRVPGM**

Only service programs that adopt the specified user profile are shown.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

**\* \_** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

#### **\*PRINT**

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

#### **\*OUTFILE**

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

Top

---

## File to receive output (OUTFILE)

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

### Qualifier 1: File to receive output

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

### Qualifier 2: Library

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

#### **\*CURLIB**

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

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

**Note:** If a new file is created, system file QADPGMAD in system library QSYS with a format name of QSYPGMAD is used as a model.

Top

---

## Output member options (OUTMBR)

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

### Element 1: Member to receive output

#### \*FIRST

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

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

### Element 2: Replace or add records

#### \*REPLACE

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

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

Top

---

## Examples

```
DSPPGMADP  USRPRF(ABC)  OUTPUT(*PRINT)
```

This command prints a list of the objects that adopt the special and private authorities of user profile ABC.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2204

User profile &1 not found.

#### CPF2213

Not able to allocate user profile &1.

#### CPF2217

Not authorized to user profile &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.

**CPF9860**

Error occurred during output file processing.

[Top](#)

---

## Display Program References (DSPPGMREF)

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

Parameters  
Examples  
Error messages

The Display Program References (DSPPGMREF) command provides a list of the system objects referred to by the specified programs. The following list shows the system objects provided for the respective program types:

### **BASIC**

\*FILE (externally described) and \*PGM

**C** no information is provided for C-language programs except for ILE C-language programs (CLE)

### **CBLLE**

\*FILE, \*PGM, and \*SRVPGM

**CL** \*FILE, \*PGM, and \*DTAARA

**CLE** \*SRVPGM

**CLLE** \*FILE, \*PGM, \*DTAARA, and \*SRVPGM

### **COBOL**

\*FILE and \*PGM (literal names on CALL command)

**CSP** \*FILE, \*PGM, \*MSGF, \*CSPMAP, and \*CSPTBL

### **PASCAL**

no information is provided for programs in PASCAL

**PL/I** \*FILE and \*PGM

**RPG** \*FILE, \*DTAARA, and \*PGM

### **RPGLE**

\*FILE, \*PGM, \*DTAARA, and \*SRVPGM

This information can be displayed, printed, or placed in a database output file.

If the information is shown or printed, a list (by library) of the specified user-authorized programs, along with the objects referenced by each program, is created. For files, information about how each file is used (input, output, update, unspecified, or any combination of these four) is also shown or printed.

If the information is written to a database file, the database file will have a record format named QWHDRPPR. The fields in record format QWHDRPPR are the same as the fields in the IBM-supplied format QWHDRPPR in file QADSPPGM in the library QSYS. The following information is contained in the database file:

- The name of the program and its text description
- The name of the library containing the program
- The number of objects referenced by the program
- The qualified name of the system object
- The information retrieval dates
- The object type of the referenced object

For files, the record contains the following additional fields:

- The name of the file in the program (possibly different from the system object name if an override was in effect when the program was created)
- The program use of the file (1=input, 2=output, 4=update, 8=unspecified, or a number representing a combination of any of these four; for example, a code of 11 is a combination of 1, 2, and 8, which is input, output, and unspecified)
- The number of record formats referenced, if any
- The name of the record format used by the file and its record format level identifier
- The number of fields referenced for each format

**Note:** This command lists which objects are referenced when the object is created or updated using UPDPGM or UPDSRVPGM. The referenced object names and libraries listed may be different than the actual names of the objects, since this information is stored when the program is created. Entries can be added as the ILE program or service program is updated using UPDPGM or UPDSRVPGM, but entries are never removed. If the object has been moved since the program was created, or an override was in effect during creation, the names listed may differ from the actual names.

**Restrictions:**

1. The user must have object operational authority for the program.
2. Also, of the libraries specified by the library qualifier, only the libraries for which the user has read authority are searched for the programs.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PGM	Program	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Program	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALLUSR, *ALL</i>	
OUTPUT	Output	<i>*, *PRINT, *OUTFILE</i>	Optional, Positional 2
OBJTYPE	Object type	Single values: *ALL Other values (up to 4 repetitions): <u>*PGM</u> , *SQLPKG, *SRVPGM, *MODULE	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

---

## Program (PGM)

Specifies the name and library of the programs whose information is shown.

This is a required parameter.

The possible values for program name are:

***program-name***

Specify the full name of a program. Information is shown only for the specified program.

***generic\*-program-name***

Specify a generic program name. Information is shown for all programs whose names begin with the specified characters.

**\*ALL** The information is shown for all programs in the library or libraries.

### Qualifier 2: Library

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

**\*CURLIB**

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

**\*USRLIBL**

If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched.

**\*ALLUSR**

All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

```
#CGULIB    #DSULIB    #SEULIB
#COBLIB    #RPGLIB
#DFULIB    #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered user libraries and are also searched:

```
QDSNX      QRCLxxxxx  QUSRIJS    QUSRVxRxMx
QGPL       QSRVAGT    QUSRINFSKR
QGPL38     QSYS2      QUSRNOTES
QMGTC      QSYS2xxxxx QUSROND
QMGTC2     QS36F      QUSRPOSGS
QMPGDATA   QUSER38    QUSRPOSSA
QMOMDATA   QUSRADSM   QUSRPYMSVR
QMOMPROC   QUSRBRM    QUSRDRARS
QPFRDATA   QUSRDIRCL  QUSRSYS
QRCL       QUSRDIRDB  QUSRVI
```

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

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

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

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_ The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

### \*PRINT

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

### \*OUTFILE

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

Top

---

## Object type (OBJTYPE)

Specifies the object type for which information is displayed.

The possible values are:

\***PGM** Only program information is displayed.

\***ALL** Program information and SQL package information are displayed.

### \*SQLPKG

Only SQL package information is displayed.

### \*SRVPGM

Service program information is displayed.

### \*MODULE

Module information is displayed.

Top

---

## File to receive output (OUTFILE)

Specifies the name and library of the database output file to which the output of the command is directed. If the file does not exist, this command creates a database output file in the specified library.

The possible library values are:

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

### \*CURLIB

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

### *library-name*

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

**Note:** The outfile format must be the same as QWHDRPPR of the system file QADSPPGM. More information on the OUTFILE format is in the Database information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter> book.

Top



---

## Output member options (OUTMBR)

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

The possible values are:

### \*FIRST

The first member in the file receives the output. If no members exist in the file, the system creates a member with the name of the file specified in the **File to receive output** prompt (OUTFILE parameter).

### *member-name*

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

The possible values for how information is stored are:

### \*REPLACE

The output data replaces any existing records in the specified file member.

**\*ADD** The output data is added to the end of existing records in the specified file member.

Top

---

## Examples

### Example 1: Storing a List of Programs

```
DSPPGMREF  PGM(LIBRARY1/*ALL)  OUTPUT(*OUTFILE)
            OUTFILE(LIB2/FILE2)
```

This command creates a list of all authorized programs found in LIBRARY1, and of the files and other system objects that the programs reference. It stores the list in a database file named FILE2 in LIB2.

### Example 2: Printing a List of Objects

```
DSPPGMREF  PGM(LIBRARY1/BILLING)  OUTPUT(*PRINT)
```

This command creates a list of system objects that are referenced by the BILLING program in LIBRARY1. The output is spooled for printing.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF3033

Object &1 in library &2 not found.

#### CPF3034

Object &1 in library &2 not displayed.

#### CPF3052

Description for file &1 not available.

#### CPF3061

Record format &3 not found for outfile &1.

#### CPF3063

Output file &1 in &2 not physical file.

- CPF3064**  
Library &1 not found.
- CPF3066**  
Error creating output file &1 in &2.
- CPF3067**  
Error while opening file &1 in &2.
- CPF3068**  
Error while writing to file &1 in &2.
- CPF3069**  
Error while closing file &1 in &2.
- CPF3070**  
Error creating member &3 in file &1.
- CPF3072**  
File &1 in &2 is a system file.
- CPF3074**  
Not authorized to library &1.
- CPF3075**  
Library &1 not available.
- CPF3076**  
Error occurred when on display.
- CPF3077**  
Error occurred when canceling display.
- CPF3084**  
Error clearing member &3 in file &1.

Top

---

## Display Program Variable (DSPPGMVAR)

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

Parameters  
Examples  
Error messages

The Display Program Variable (DSPPGMVAR) command shows the current value of one or more program variables in a program that is being debugged. The variables can be specified either by their variable names or by their machine-instruction object-definition-table-vector (MI ODV) numbers. A maximum of 10 variables can be specified.

### Restrictions:

- You can use this command only in debug mode. To start debug mode, refer to the Start Debug (STRDBG) command.
- You cannot use this command if you are servicing another job, and that job is on a job queue, or is being held, suspended, or ended.
- You cannot use this command to display variables in a bound program.
- You cannot use this command to display variables within the system domain unless the user has \*SERVICE special authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PGMVAR	Program variables	Values (up to 10 repetitions): <i>Element list</i>	Required, Positional 1
	Element 1: Program variable	<i>Character value</i> , *CHAR	
	Element 2: Basing pointer variable	Values (up to 5 repetitions): <i>Character value</i>	
OUTFMT	Output format	*CHAR, *HEX	Optional
OUTPUT	Output	*, *PRINT	Optional
PGM	Program	<i>Name</i> , *DFTPGM	Optional
START	Char output start position	<i>Integer</i> , <u>1</u>	Optional, Positional 2
LEN	Characters to display	<i>Integer</i> , *DCL	Optional
RCRLVL	Recursion level	<i>Integer</i> , *LAST	Optional

Top

---

## Program variables (PGMVAR)

Specifies up to ten program variables whose values are to be shown. The variables can be in a high-level language (HLL) or machine instruction (MI) program.

This is a required parameter.

### Element 1: Program variable

### **\*CHAR**

This special value can be specified instead of a variable name if a basing pointer is specified. This shows a character view of a pointer to be shown without the use of a based variable.

#### *character-value*

Specify the name of the program variable to be shown. The name must be enclosed in apostrophes if it contains special characters.

If the program variable is an array, the subscripts representing the elements in the array can be specified. If an array name is specified without any subscripts, all of the array elements are recorded. A single-dimensional cross-section can also be specified. Up to 132 characters may be specified for this program variable entry. This includes any qualifiers, subscripts, blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, machine-interface object-definition-table-vector (MI ODV) number, asterisk (single-dimensional cross-section), or a numeric variable name can be specified for a subscript.

### **Element 2: Basing pointer variable**

#### *character-value*

Specify the name of a basing pointer variable. In some languages, the program variable may be based on a pointer variable. This set of values allows you to explicitly specify the basing pointers for the variable to be recorded. Each basing pointer name must be enclosed in apostrophes if it contains special characters.

If the basing pointer is an array, the subscripts representing an element in the array must be specified. Up to 132 characters can be specified for a basing pointer name. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, MI ODV number, or a numeric variable name can be specified for a subscript.

Top

---

## **Output format (OUTFMT)**

Specifies the format in which the values of the program variables are shown. Additional descriptive information for some variable types is also shown with the variable values in a format predefined by the system.

### **\*CHAR**

The program variables are shown in character form.

**\*HEX** The program variables are shown in hexadecimal form.

Top

---

## **Output (OUTPUT)**

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

**\*** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

### **\*PRINT**

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

Top

---

## Program (PGM)

Specifies the name of the program that contains the program variables shown.

### \*DFTPGM

The program previously specified as the default program has its variables shown.

*name* Specify the name of the program whose program variables are shown.

Top

---

## Char output start position (START)

Specifies, for string variables only, the starting position in the string from which the value of the variable is to be shown. If more than one string variable is specified for the **Program variables (PGMVAR)** parameter, the same starting position value is used for each one. For a bit string, the value specifies the starting bit position; for a character string, the value specifies the starting character position.

1 The variable is shown from the first position on through the length specified for the **Characters to display (LEN)** parameter.

### *integer*

Specify the first position in the string to be shown. The value supplied for the **Char output start position (START)** parameter must not be larger than the maximum string length for any variable specified, except that a value of **1** for the START parameter is allowed if the maximum length for a string is zero. The value supplied for the LEN parameter plus the value supplied for the START parameter minus one must not be greater than the maximum string length. These checks are made for each string variable specified for the PGMVAR parameter.

Top

---

## Characters to display (LEN)

Specifies, for string variables only, the length of the string shown when the breakpoint is reached, starting at the position specified by the **Char output start position (START)** parameter. If more than one string variable is specified for the **Program variables (PGMVAR)** parameter, the same value is used for each one. For a bit string, the value specifies the number of bits shown, and for a character string, the value specifies the number of characters shown.

\*DCL The string variable is shown to the end of the string or for a value of 200 bytes, whichever is less. If the string variable has a maximum length of zero, the only allowable value for the LEN parameter is \*DCL.

### *integer*

Specify the length of the data shown. The length (as well as the combination of values supplied for the START parameter and the LEN parameter must be no greater than the length of the shortest string specified by the PGMVAR parameter.

Top

---

## Recursion level (RCRLVL)

Specifies which recursion level of the program contains the variable whose value is to be displayed. Recursion level 1 is the first (or earliest) call of the program, recursion level 2 is the second call of the program, and so on to the last (most recent) recursion level in the stack. For example, if program A calls program B, then program B calls program A, a new recursion level of program A is formed. If the first

call of program A contains the variable to be displayed, a value of 1 for the **Recursion level (RCRLVL)** parameter must be specified. Some high-level languages also allow recursive procedures.

**\*LAST**

The value of the specified variable in the last (most recent) call of the specified program is displayed.

***integer***

Specify the recursion level of the program that contains the variable whose value is to be displayed.

Top

---

## Examples

### Example 1: Displaying Program Variables

```
DSPPGMVAR PGMVAR('&QUANT') PGM(MYPROG)
```

Assuming that the program MYPROG is in debug mode, this command shows the name and current value of the CL variable called &QUANT its type and length are also shown.

### Example 2: Displaying Program Variables

```
DSPPGMVAR PGMVAR(TOTSALES MANHRS) PGM(REGION) RCRLVL(1)
```

This command shows the program variables TOTSALES and MANHRS of the first call of the program REGION.

Top

---

## Error messages

### **\*ESCAPE Messages**

#### **CPF1999**

Errors occurred on command.

Top

## Display Problems (DSPPRB)

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

Parameters  
 Examples  
 Error messages

The Display Problem (DSPPRB) command allows you to display or print service information related to performing hardware or software maintenance. The service information, contained in the problem log entries, are shown on the DSPPRB display, printed with the job's output, or stored in a database file.

Top

### Parameters

Keyword	Description	Choices	Notes
PRBID	Problem identifier	Character value, <u>*ALL</u>	Optional, Positional 1
STATUS	Status type	Single values: <u>*ALL</u> Other values (up to 6 repetitions): *OPENED, *READY, *PREPARED, *SENT, *ANSWERED, *VERIFIED, *CLOSED	Optional
SEV	Severity	Single values: <u>*ALL</u> Other values (up to 3 repetitions): 1, 2, 3, 4	Optional
PERIOD	Period	Element list	Optional
	Element 1: Start time and date	Element list	
	Element 1: Start time	Time, <u>*AVAIL</u>	
	Element 2: Start date	Date, <u>*BEGIN</u> , *CURRENT	
	Element 2: End time and date	Element list	
	Element 1: End time	Time, <u>*AVAIL</u>	
	Element 2: End date	Date, <u>*END</u> , *CURRENT	
HARDWARE	Hardware	Element list	Optional
	Element 1: Device type	Character value, <u>*ALL</u>	
	Element 2: Model number	Character value, <u>*ALL</u>	
	Element 3: Serial number	Character value, <u>*ALL</u>	
RESOURCE	Resource name	Name, <u>*ALL</u>	Optional
LICPGM	Product	Element list	Optional
	Element 1: Program identifier	Character value, <u>*ALL</u>	
	Element 2: Release	Character value, <u>*ALL</u>	
	Element 3: Modification level	Character value, <u>*ALL</u>	
FUNCTION	Function	Character value, <u>*ALL</u>	Optional
PGM	Program	Generic name, name, <u>*ALL</u>	Optional
MSGID	Message identifier	Generic name, name, <u>*ALL</u>	Optional

Keyword	Description	Choices	Notes
ORIGIN	Origin	<i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *ALL, *NETATR</i>	
	Element 2: Control point name	<i>Communications name, *ALL, *NETATR</i>	
SRVID	Service number	<i>Character value, *ALL</i>	Optional
ASNUSER	User assigned	<i>Simple name, *ALL</i>	Optional
GROUP	Group assigned	<i>Character value, *ALL</i>	Optional
PRBTYPE	Problem type	<i>*ALL, 1, 2, 3, 4, 5, 6</i>	Optional
PRBCGY	Problem category	<i>*ALL, *REPORT, *CRITICAL, *LOGONLY</i>	Optional
OUTPUT	Output	<i>_, *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>	
TYPE	Type of information	<i>*BASIC, *CAUSE, *FIX, *USRTXT, *SPTDTA</i>	Optional

Top

---

## Problem identifier (PRBID)

Specifies the ID of the problems to be selected. Problems with different origin systems may have the same identifier. Specifying the **Origin (ORIGIN)** parameter along with this parameter will assure that only one problem is selected.

**\*ALL** All problems that match the other selection criteria are selected.

### *problem-ID*

Specify the 10-character problem ID of the problem to be selected.

Top

---

## Status type (STATUS)

Specifies the status of problem log entries. There are seven types of status:

### **\*OPENED**

The problem is in Opened status. The problem has been identified and a problem record was created.

### **\*READY**

The problem is in Ready status. Problem analysis information has been added to the problem record.

### **\*PREPARED**

The problem is in Prepared status. The problem has been prepared for reporting.



**\*SENT**

The problem is in Sent status. The problem has been sent to a service provider, but no answer has been returned.

**\*ANSWERED**

The problem is in Answered status. An answer has been returned by the service provider or added by an operator on this system.

**\*VERIFIED**

The problem is in Verified status. The problem was resolved and the system operator has verified that the problem is corrected.

**\*CLOSED**

The problem was closed.

**\*ALL** All types of status of problem log entries are shown.

***status-type***

Specify the type of status to be shown.

Top

---

## Severity (SEV)

Specifies the severity level of the problem. Severity levels are assigned by the user when the problem is prepared for reporting. The four severity levels are:

- 1 High
- 2 Medium
- 3 Low
- 4 None

**\*ALL** All problem log entries are shown.

***severity***

Specify the level of severity of problem log entries to be shown.

Top

---

## Period (PERIOD)

Specifies the period of time for which the problem data is selected. The dates and times indicate when the problem log entry was created.

The possible **starting time** values are:

**\*AVAIL**

Problem entries created on the specified starting date are shown.

***start-time***

Specify the creation time (for the specified date) of the first data record to include.

The time is specified in 24-hour format and 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. Hours, minutes, and seconds must each be exactly 2 digits (use leading zeros if necessary).
- 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.

The possible **starting date** values are:

**\*BEGIN**

Problems available at the beginning of the log are shown. If \*BEGIN is specified, time values other than \*AVAIL for start-time are ignored.

**\*CURRENT**

The problem log entries created for the current day between the specified start-time and end-time (if specified) are displayed.

***start-date***

Specify the date after which the problem log entries can be shown. The date must be specified in the job date format.

The possible **ending time** values are:

**\*AVAIL**

Problem log entries created on the specified ending date are shown.

***end-time***

Specify the time after which problem log entries created are not shown. See the start-time description in this parameter description for details on how the time must be specified.

The possible **ending date** values are:

**\*END** The last day on which a problem log entry was created is the last day for which problems are shown. If \*END is specified, then any time value other than \*AVAIL for end-time is ignored.

**\*CURRENT**

The current date is used for the date of the last problem opened to show.

***end-date***

Specify the date after which problem log entries cannot be shown. See the start-date description in this parameter for details on how the date must be specified.

Top

---

## Hardware (HARDWARE)

Specifies that only problem log entries that identify the specified failing hardware are shown.

The possible **machine type** values are:

**\*ALL** All entries are shown, regardless of what hardware is identified as failing.

**Note:** If \*ALL is specified, any value other than \*ALL for model and serial number is ignored.

***type*** Specify the 4-character type code for the hardware.

The possible **model number** values are:

**\*ALL** All entries that identify failing hardware of the specified type are shown.

**Note:** If \*ALL is specified, any value other than \*ALL for serial number is ignored.

***model*** Specify the 3-character model number for the hardware.

The possible **serial number** values are:

**\*ALL** All entries that identify failing hardware of the specified type and model are shown.

*serial* Specify the serial number of the hardware in one of the following formats, where n is a decimal character ranging from 0 through 9.

- nnnnn
- nnnnnnn
- nn-nnnnn
- nn-nnnnnnn

Top

---

## Resource name (RESOURCE)

Specifies that only problem log entries that identify the specified failing resource name are shown.

**\*ALL** All entries are shown, regardless of which resource name, if any, is identified by the problem.

*resource-name*

Specify the resource name.

Top

---

## Product (LICPGM)

Specifies that problem log entries that identify the specified failing software are shown.

The possible **program ID** values are:

**\*ALL** All entries are shown regardless of what software is identified as failing.

**Note:** If \*ALL is specified, any value other than \*ALL for release and modification is ignored.

*program-ID*

Specify the identification number of the licensed program.

The possible **release** values are:

**\*ALL** All entries that identify failing software from the specified licensed program are shown.

**Note:** If \*ALL is specified, any value other than \*ALL for modification is ignored.

*release-level*

Specify the release level of the licensed program.

The possible **modification** values are:

**\*ALL** All entries that identify failing software from the specified licensed program and release are shown.

*modification*

Specify the modification number of the release.

Top

---

## Function (FUNCTION)

Specifies that only problem log entries that identify the specified function are to be shown. The function is present only in user-detected problem log entries and represents a major functional area or product.

**\*ALL** All entries are shown regardless of what function ID, if any, is identified.

*generic\*-function-identifier*

Specify a generic function ID. A generic ID is a character string containing one or more of the initial characters of the function ID followed by an asterisk (\*). If blank characters are included, the character string must be enclosed in apostrophes (').

*function-id*

Specify the complete function ID. If blank characters are included, the character string must be enclosed in apostrophes (').

The possible values for function ID are:

<b>ID</b>	<b>Description</b>
<b>ALRT</b>	Alerts
<b>APF</b>	Advanced printer function
<b>APPC</b>	Advanced program-to-program communications
<b>APPN</b>	Advanced Peer-to-Peer Networking function
<b>APS</b>	Advanced DBCS Printer Support program
<b>ASYNC</b>	Asynchronous communications
<b>BAS</b>	AS/400 BASIC
<b>BGU</b>	AS/400 Business Graphics Utility
<b>BSC</b>	Binary synchronous communications
<b>CBL</b>	AS/400 COBOL and COBOL for iSeries
<b>CGU</b>	Character generator utility
<b>CL</b>	Control language
<b>CRYPT</b>	Cryptographic Support/400
<b>CSM</b>	Communications and Systems Management
<b>C400</b>	C/400
<b>DDM</b>	Distributed data management
<b>DFU</b>	Data file utility
<b>DHCF</b>	Distributed host command facility
<b>DICT</b>	Language Dictionaries/400
<b>DSNX</b>	Distributed system node executive
<b>ECS</b>	Electronic customer support
<b>FINANCE</b>	Finance Communications
<b>FTN</b>	FORTTRAN/400
<b>FTS</b>	File transfer support
<b>GDDM</b>	Graphical Data Display Manager
<b>ICF</b>	Intersystem Communication Facility
<b>IDU</b>	Interactive data definition utility (IDDU)

**INTRA** Intrasystem Communication

**IPDS** Intelligent Printer Data Stream

**ITF** Interactive terminal facility

**LPDA** Link Problem Determination and Analysis

**MIA** MI Assembler

**MIGR** Migration

**OBJD** Object distribution

**OCL** S/36 operator control language

**OFC** OfficeVision

**OSP** i5/OS

**PAS** AS/400 Pascal

**PASSTHRU** Pass-through

**PC400** Client Access/400

**PDM** Programming development manager

**PL1** AS/400 PL/I

**POSCOM** Point-of-Sale Communications Utility/400

**QRY** Query/400

**RETAIL** Retail Communications

**REXX** REXX/400

**RJE** Remote job entry

**RLU** Report layout utility

**RPFT** Performance Tools/400

**RPG** RPG for iSeries

**RSCS** Remote spooling communications subsystem

**RWS** Remote work station

**SDA** Screen design aid

**SDLC** Synchronous Data Link Control

**SEU** Source entry utility

**SMU** IBM System Manager for i5/OS

**SNADS** SNA distribution services

**SNUF** SNA upline facility

**SORT** AS/400 sort

**SQL** SQL/400

**SUU** System upgrade utility

TCPIP TCP/IP Connectivity Utilities/400  
TRLAN Token ring network  
TXT38 System/38 compatible Text Management  
VMC Vertical Licensed Internal Code  
VNET RSCS/PROFS  
WRD Word processing  
X21 X.21 public data network  
X25 X.25 packet-switching data network  
3270EM 3270 device emulation  
802.3 Ethernet

Top

---

## Program (PGM)

Specifies whether only problem log entries that identify a specified failing program are shown. For machine detected problems, the failing program, if any, is identified by the most likely possible cause.

**\*ALL** All entries are shown regardless of whether a program is identified.

*generic\*-program-name*

Specify a generic program name. A generic name is a character string containing one or more characters followed by an asterisk (\*).

*program-name*

Specify the program name.

Top

---

## Message identifier (MSGID)

Specifies whether only problem log entries that identify a specified message are shown.

**\*ALL** All entries are shown regardless of which message ID is associated with the problem.

*generic\*-msg-id*

Specify a generic message identifier (ID). A generic ID is a character string containing one or more characters followed by an asterisk (\*).

*message-identifier*

Specify the message identifier.

Top

---

## Origin (ORIGIN)

Specifies that only problem log entries that originated at the specified systems are shown.

The possible **network ID** values are:

**\*ALL** All entries are shown regardless of the network ID of the origin system.

**\*NETATR**

Only entries that originated on systems with the same local network ID as the one defined in the network attributes for this system are shown.

*network-ID*

Specify a network ID. Only entries that originated on systems with this local network ID are shown.

The possible **control point name** values are:

**\*ALL** All entries originating at systems with the specified network ID are shown.

**\*NETATR**

Only entries that originated on systems with the same local control point name as the one defined in the network attributes for this system are shown.

*control-point-name*

Specify a control point name.

Top

---

## Service number (SRVID)

Specifies whether only problem log entries that have the specified service assigned number are shown. This number was assigned when the problem was reported to IBM service support.

**\*ALL** All entries are shown regardless of the service assigned number, if any.

*service-assigned-number*

Specify the service assigned number.

Top

---

## User assigned (ASNUSER)

Specifies whether only problem log entries assigned to this user are shown.

**\*ALL** All problem log entries are shown.

*user-name*

Specify the user ID assigned to the problems to be shown.

Top

---

## Group assigned (GROUP)

Specifies the group in the filter to which the problem is assigned.

**\*ALL** All problem log entries are shown, regardless of the group assigned to them.

*group-name*

Specify the 10-character problem filter group assigned to the entry.

**Note:** The values are blank if problem log filtering is not used.

Top

---

## Problem type (PRBTYPE)

Specifies which type of problems to display.

**\*ALL** All problem log entries are shown, regardless of the problem type.

- 1 Only machine-detected problems are shown.
- 2 Only user-detected problems are shown.
- 3 Only PTF order problems are shown.
- 4 Only application-detected problems are shown.
- 5 Only Client machine-detected problems are shown.
- 6 Only Client user-detected problems are shown.

Top

---

## Problem category (PRBCGY)

Specifies which category of problems to display.

**\*ALL** All problems are shown.

**\*REPORT**

Problems that are logged and reported to the service provider are shown.

**\*CRITICAL**

Problems that are critical are shown.

**\*LOGONLY**

Problems that are logged, but not reported to the service provider, are shown.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

**\* \_** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

**\*PRINT**

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

**\*OUTFILE**

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

Top



---

## File to receive output (OUTFILE)

Specifies the name and library of the database file where the output of the command is directed. If the output file does not exist, this command creates a database file in the specified library. If a file is created, the text says **OUTFILE for DSPPRB**, and the authority for users other than those who have specific authority, group authority, or authority from an authorization list 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.

The possible library values are:

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

**\*CURLIB**

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

*library-name*

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

Top

---

## Output member options (OUTMBR)

Specifies the name of the database file member to which the output of the command is directed and whether the output to that member replaces or is added to existing records.

The possible methods for adding new records are:

**\*FIRST**

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

*member-name*

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

The optional values are:

**\*REPLACE**

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

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

**Note:** The **OUTFILE** and **OUTMBR** keywords are conditioned on **OUTPUT(\*OUTFILE)**.

If **OUTPUT(\*OUTFILE)** is specified, a database file name is required.

Top

---

## Type of information (TYPE)

Specifies the type of problem information that is placed in the output file. Only one type of data can be placed in each file. This parameter is valid only when a value is also specified on the **File to receive output (OUTFILE)** parameter.

### **\*BASIC**

Basic problem data including problem type, and status; device type, model, and serial number; product ID; contact information; and tracking data is placed in the output file.

### **\*CAUSE**

Possible problem causes are placed in the output file.

**\*FIX** Program temporary fixes (PTFs) are placed in the output file.

### **\*USRTEXT**

User-supplied text (note records) is placed in the output files.

### **\*SPTDTA**

Supporting data identifiers are placed in the output file.

Top

---

## Examples

### **Example 1: Displaying Today's Problem Log Entries**

```
DSPPRB PERIOD>(*AVAIL *CURRENT) (*AVAIL *CURRENT)
```

This command shows all problem log entries that were created today.

### **Example 2: Creating an Output File**

```
DSPPRB OUTPUT(*OUTFILE) OUTFILE(*CURLIB/NEWFILE)
```

This command creates a member in the physical file NEWFILE in the current library which contains a record for each problem log entry in the problem log.

### **Example 3: Displaying a List of Hardware Problems**

```
DSPPRB SEV(1 2) HARDWARE(9347 001 10-7523489)
```

This command shows a list containing problems with the hardware specified by the user. The user has specified that the command track medium-to-high levels of severity.

### **Example 4: Displaying a List of Problems That Have Been Opened**

```
DSPPRB STATUS(*OPENED)
PERIOD(*AVAIL *CURRENT) (120000 *CURRENT)
LICPGM(5716SS1 03 00) PGM(QNOPGM)
```

This command shows a list containing problems that have been opened during the period starting at midnight and ending at noon on the current day, and have not yet been analyzed. This command also identifies the specified licensed program identifier and program name as the probable cause of the failure.

### **Example 5: Displaying a List of Machine-Detected Problems**

```
DSPPRB RESOURCE(TAP01) MSGID(CPF6788)
```

This command shows a list containing machine-detected problems that were opened due to the message, CPF6788, having been sent to the system operator message queue. The list of problems includes

user-detected problems. To get the user-detected problems, the user specified the resource name and message identifier by using the Analyze Problem (ANZPRB) command.

#### **Example 6: Displaying a List of Reported Problems**

```
DSPPRB  SRVID(12345)
```

This command shows a list containing problems that have been reported to an IBM service support center and have 12345 as the service identifier.

[Top](#)

---

## **Error messages**

### **\*ESCAPE Messages**

#### **CPF7A9C**

Cannot work with the problem log at this time.

#### **CPF9845**

Error occurred while opening file &1.

#### **CPF9847**

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

[Top](#)



---

## Display PSF Configuration (DSPPSF CFG)

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

Parameters  
Examples  
Error messages

The Display PSF Configuration (DSPPSF CFG) command displays a Print Services Facility (PSF) configuration object from the specified library.

### Restrictions:

- The PSF feature is required to use this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PSFCFG	PSF configuration	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: PSF configuration	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTPUT	Output	<i>*, *PRINT _</i>	Optional, Positional 2

Top

---

## PSF configuration (PSFCFG)

Specifies the Print Services Facility (PSF) configuration object to display.

This is a required parameter.

### Qualifier 1: PSF configuration

*name* Specify the name of the PSF configuration object to display.

### Qualifier 2: Library

**\*LIBL** Search all libraries in the job's library list until the first match is found.

### \*CURLIB

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

*name* Search the specified library.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting workstation or printed with the job's spooled output.

\*  
\_ The output is displayed if requested by an interactive job and printed with the job's spooled output if requested by a batch job.

**\*PRINT**

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

Top

---

## Examples

```
DSPPSF CFG PSFCFG(PSF CFG1) OUTPUT(*PRINT)
```

This command will print a description of the Print Services Facility (PSF) configuration object (\*PSFCFG) named PSFCFG1. The libraries in the job's library list will be searched for the PSFCFG1 configuration object.

Top

---

## Error messages

### \*ESCAPE Messages

**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.

Top

---

## Display Program Temporary Fix (DSPPTF)

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

Parameters  
Examples  
Error messages

The Display Program Temporary Fix (DSPPTF) command shows the program temporary fixes (PTFs) for a specified product.

**Restriction:** This command is shipped with public \*EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
LICPGM	Product	Character value, <u>*ALL</u>	Optional, Positional 1
SELECT	PTF numbers to select	Character value, <u>*ALL</u> , *PTFSAVE, *ONORDER, *ACTRQD	Optional, Positional 2
RLS	Release	Character value, <u>*ALL</u>	Optional
COVERONLY	Cover letter only	<u>*NO</u> , *YES	Optional
OUTPUT	Output	<u>*</u> , *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	Qualified object name	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
OUTMBR	Output member options	Element list	Optional
	Element 1: Member to receive output	Name, <u>*FIRST</u>	
	Element 2: Replace or add records	<u>*REPLACE</u> , *ADD	

Top

---

### Product (LICPGM)

Specifies the product for the PTFs that are shown. When LICPGM(\*ALL) is specified, PTFs for all installed and supported products are shown.

The possible values are:

\*ALL The PTFs of all installed and supported products are shown.

*licensed-program*

Specify the product for which PTFs are shown.

Top

---

## PTF numbers to select (SELECT)

Specifies which PTF is shown for the specified product. \*ALL cannot be specified for the **Product (LICPGM)** parameter if a PTF number is specified on the **PTF numbers to select (SELECT)** parameter.

When LICPGM(\*ALL) and SELECT(\*ALL) are specified, all PTFs for all installed and supported products are shown.

When LICPGM(licensed-program) and SELECT(\*ALL) are specified, PTFs for all releases of the specified products are shown. The possible values are:

**\*ALL** The status of all PTFs for the specified product is shown.

### **\*PTFSAVF**

The status of all PTFs for the specified product that have a save file in library QGPL is shown. This special value is useful for determining which PTF save files are no longer needed. It is also useful for the service provider for determining which PTFs can be distributed.

### **\*ONORDER**

The status of all PTFs for the specified product that are on order is shown.

### **\*ACTRQD**

The PTFs that have required actions pending are shown. This special value is useful to the user for determining which PTFs require an action to become active.

**Note:** If \*ACTRQD is specified, the exit programs that run take a long time, resulting in a delay in the appearance of the first screen.

### ***PTF-number***

Specify the PTF identification number of the PTF that is shown.

Top

---

## Release (RLS)

Specifies the release level of the PTFs being displayed.

The possible values are:

**\*ALL** The PTFs for all releases of the supported and installed products are displayed.

### ***release-level***

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

If the release-level specified is the release-level of the installed base option of the product, PTFs for all installed options of the product are displayed regardless of the release-level of the option. All PTFs for options that are supported at this release-level are also displayed.

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

Top



---

## Cover letter only (COVERONLY)

Specifies whether only the cover letter is displayed.

The possible values are:

**\*NO** A cover letter is not displayed.

**\*YES** The cover letter is displayed.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

The possible values are:

**\*** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

**\*PRINT**

The output is printed with the job's spooled output. The name of the spooled output file is QSYSPRT.

**\*OUTFILE**

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

Top

---

## File to receive output (OUTFILE)

Specifies the physical database file where the PTF records are directed. If the output file already exists, the system attempts to use it. The system creates a physical database file (with the name specified in the OUTFILE parameter) in the designated library. A member is created for the file with the name specified in the **Output member options (OUTMBR)** parameter. If a new file is created, system file QADSPPTF in system library QSYS with a format name of QSCPTF is used as a model.

The possible library values are:

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

**\*CURLIB**

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

**library-name**

Specify the library where the file is located.

The possible values are:

**file-name**

Specify the name of the file that receives the PTFs.

Top

---

## Output member options (OUTMBR)

Specifies the name of the database file member where the output of the command is directed. A second value specifies whether the new data replaces the existing data or is added to the end of the data already in the file member.

The possible member name values are:

### \*FIRST

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

### *member-name*

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

The possible placement values are:

### \*REPLACE

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

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

Top

---

## Examples

### Example 1: Printing Status of PTFs

```
DSPPTF LICPGM(5722SS1) OUTPUT(*PRINT)
```

This command produces a printout containing the status of PTFs for the product 5722SS1.

### Example 2: Printing Information

```
DSPPTF LICPGM(5722SS1) SELECT(SI00034) OUTPUT(*PRINT)
```

This command produces a printout containing detailed information about PTF SI00034 for the product 5722SS1.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF0C4B

Product availability object &2/&1 recovery required.

#### CPF0C4C

Cannot allocate object &1 in library &2.

#### CPF0C4D

Error occurred while processing object &1 in library &2.

#### CPF24B4

Severe error while addressing parameter list.

**CPF35BE**  
Product &1 &3 not supported or installed.

**CPF35F5**  
Cover letter not found for PTF &1-&2 &3.

**CPF358A**  
Release not valid.

**CPF3925**  
Cannot open file &1.

**CPF3950**  
Error message &2 received for file &1. Request ended.

**CPF6601**  
No PTF activity exists for product &1.

**CPF6602**  
PTF &1-&2 &3 not found.

**CPF6603**  
No PTFs found.

**CPF6613**  
No PTFs met selection criteria.

**CPF8191**  
Product definition &4 in &9 damaged.

**CPF8193**  
Product load object &4 in &9 damaged.

**CPF9860**  
Error occurred during output file processing.

Top



---

## Display PTF Cover Letter (DSPPTFCVR)

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

Parameters  
Examples  
Error messages

The Display Program Temporary Fix Cover Letter (DSPPTFCVR) command shows the program temporary fix (PTF) cover letters for a specified product.

**Restriction:** This command is shipped with public \*EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
LICPGM	Product	Character value, <u>*ALL</u>	Optional, Positional 1
SELECT	PTF numbers to select	Values (up to 300 repetitions): Character value, <u>*NOTAPY</u> , <u>*ALL</u>	Optional, Positional 2
RLS	Release	Character value, <u>*ALL</u>	Optional
CVRATR	Attributes	Single values: <u>*ALL</u> Other values (up to 7 repetitions): *SPCINST, *PREIMM, *PREDLY, *PREOPR, *POSTOPR, *UNKNOWN	Optional
CVRLRLNG	Cover letter language	Character value, <u>*DFT</u> , <u>*ALL</u>	Optional
OUTPUT	Output	<u>_</u> , *PRINT	Optional

Top

---

### Product (LICPGM)

Specifies the product for which cover letters are displayed.

The possible values are:

\*ALL The cover letters for all installed or supported products are displayed.

*licensed-program*

Specify the product for which cover letters are displayed.

Top

---

### PTF numbers to select (SELECT)

Specifies which cover letter is displayed for the specified product.

\*ALL cannot be specified for the **Product (LICPGM)** parameter if a PTF number is specified on the **PTF numbers to select (SELECT)** parameter.

When LICPGM(\*ALL) and SELECT(\*ALL) are specified, cover letters for all installed or supported products are displayed.

When LICPGM(licensed-program) and SELECT(\*ALL) are specified, cover letters for all releases of the specified product are displayed. The possible values are:

**\*NOTAPY**

All cover letters of the PTFs that are not already applied or superseded for the specified product are displayed.

**\*ALL** All cover letters for the specified product are displayed.

***PTF-number***

Specify the PTF identification number of the cover letter that is displayed. A maximum of 300 PTF numbers can be specified.

Top

---

## Release (RLS)

Specifies the release level of the PTFs being displayed.

The possible values are:

**\*ALL** The cover letters for all releases of the installed or supported products are displayed.

***release-level***

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

If the release specified is the release of the installed base option of the product, cover letters for all installed options of the product are displayed regardless of the release of the option. All cover letters for options that are supported at this release are also displayed.

If the release specified is not the release of the installed base option of the product, only cover letters of the product options that are supported or installed at that release are displayed.

Top

---

## Attributes (CVRATR)

Specifies which cover letter to display based on the attributes contained in the cover letter.

**Note:** The CVRATR parameter can be specified only with SELECT(\*ALL) or SELECT(\*NOTAPY).

The possible values are:

**\*ALL** All cover letters are displayed.

**\*SPCINST**

Only cover letters with special instructions are displayed.

**\*PREIMM**

Only cover letters with special instructions that need to be followed before being applied or removed immediately are displayed.

**\*PREDLY**

Only cover letters with special instructions that need to be followed before being applied or removed during an IPL (delayed) are displayed.

**\*PREOPR**

Only cover letters with special instructions that need to be followed before being applied or removed (immediately or delayed) are displayed.

**\*POSTOPR**

Only cover letters with special instructions that need to be followed after being applied or removed are displayed.

**\*UNKNOWN**

Display the cover letters for which the system cannot detect if they have attributes. The most likely reasons are when the PTF cover letter was created prior to operating system release V5R1M0, or the cover letter was created using the System Manager licensed program.

Top

---

## Cover letter language (CVRLTRLNG)

Specifies the language feature code that is used to display the cover letters.

The possible values are:

**\*DFT** Cover letters are displayed in the default language feature code. If there is only one cover letter for the PTF, it will be displayed. If there is more than one cover letter for the PTF, the following criteria will be used to determine which cover letter to display.

The language feature code that matches the service contact information (WRKCNTINF) will be used.

If no language feature code matches the service contact information, the language feature code that matches the primary language of the operating system will be used.

If no cover letters match the language feature code in the service contact information, or the language feature code of the operating system, all cover letters for the PTF will be displayed. The user will be given a list of language feature codes to choose from (if requested by an interactive job) or all cover letters will be printed with the job's spooled output (if requested by a batch job).

**\*ALL** Cover letters for all languages are displayed.

*language-feature-code*

Cover letters for the selected language feature code are displayed. If a cover letter for the selected language feature code is not available, no cover letter is displayed for the PTF.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting workstation or printed with the job's spooled output.

The possible values are:

**\*** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

**\*PRINT**

The output is printed with the job's spooled output. The name of the spooled output file is QSYSPT.

Top

---

## Examples

### Example 1: Print PTF Cover Letters

```
DSPPTFCVR LICPGM(5722SS1) SELECT(SI00034) OUTPUT(*PRINT)
```

This command produces a printout of the cover letter for PTF SI00034 in the product 5722SS1.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF0C4B

Product availability object &2/&1 recovery required.

#### CPF0C4C

Cannot allocate object &1 in library &2.

#### CPF0C4D

Error occurred while processing object &1 in library &2.

#### CPF35BE

Product &1 &3 not supported or installed.

#### CPF35D5

Cover letter NLV not valid.

#### CPF35F5

Cover letter not found for PTF &1-&2 &3.

#### CPF358A

Release not valid.

#### CPF3586

List of PTFs not correct.

#### CPF6601

No PTF activity exists for product &1.

#### CPF6602

PTF &1-&2 &3 not found.

#### CPF6603

No PTFs found.

[Top](#)



---

## Display Power On/Off Schedule (DSPPWRSCD)

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

Parameters  
Examples  
Error messages

The Display Power On/Off Schedule (DSPPWRSCD) command allows you to display or print the power on/off schedule.

Top

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *PRINT _	Optional, Positional 1
STRDATE	Start date	<i>Date</i> , * <b>TODAY</b>	Optional
DAYS	Days	1-366, <b>40</b>	Optional

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*     The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).  
\_

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

Top

---

### Start date (STRDATE)

Specifies the first date to be displayed or printed on the power on/off schedule.

**\*TODAY**  
The current date is used.

*date*   Specify a date in the future or the current date. The date must be entered in the same format as specified by your job attributes.

Top

---

### Days (DAYS)

Specifies the number of days for which the power on/off schedule is to be printed.

**40**     Forty days of the schedule are printed.

1-366 Specify the number of days.

Top

---

## Examples

### Example 1: Displaying Power On/Off Schedule

```
DSPPWRSCD
```

This command displays the power on/off schedule.

### Example 2: Printing 30 Days of Power On/Off Schedule

```
DSPPWRSCD OUTPUT(*PRINT) DAYS(30)
```

This command prints 30 days of the power on/off schedule, starting with the current date.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1E2B

Power scheduler and cleanup options not found.

#### CPF1E23

Power schedule or cleanup options in use by another user.

#### CPF1E27

Not authorized to change power on/off schedule.

#### CPF1E28

Cannot print schedule at specified STRDATE.

#### CPF1E99

Unexpected error occurred.

Top

---

## Display Record Locks (DSPRCDLCK)

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

Parameters  
Examples  
Error messages

The Display Record Locks (DSPRCDLCK) command allows you to show the current record lock status of a particular database physical file member. This command displays the lock status for a particular relative record number, or the lock status of all locked records in the member.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FILE	Physical file	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Physical file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MBR	Member	<i>Name, *FIRST</i>	Optional, Positional 2
RCDNBR	Record number	1-4294967288, <u>*ALL</u>	Optional, Positional 3
OUTPUT	Output	<i>_, *PRINT</i>	Optional

Top

---

### Physical file (FILE)

Specifies the physical file that contains the member whose record locks are shown.

This is a required parameter.

#### Qualifier 1: Physical file

*name* Specify the name of physical file.

#### Qualifier 2: Library

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

#### \*CURLIB

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

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

Top

---

### Member (MBR)

Specifies the member in the file whose record locks are shown.

### \*FIRST

The first member of the specified file is used.

*name* Specify the name of the physical file member.

Top

---

## Record number (RCDNBR)

Specifies a particular relative record number or all records of a member.

\*ALL The lock status of all records currently locked in a physical file member is shown.

*1-4294967288*

Specify the record number of the record whose lock status is to be displayed.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\* The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).  
-

### \*PRINT

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

Top

---

## Examples

```
DSPRCDLCK FILE(MASTER/PAYROLL) MBR(*FIRST) RCDNBR(1)
          OUTPUT(*)
```

This command shows the lock status of relative record number 1, in the first member of the physical file named PAYROLL in the MASTER library.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF3130

Member &2 already in use.

#### CPF3210

File &1 in library &2 not correct type.

#### CPF3247

Record number &4 does not exist in member &3.

#### CPF3275

Member &3 file &1 in &2 not found.

- CPF7D41**  
Error occurred while logging order assistance request.
- CPF7D42**  
Error occurred while performing database operation.
- CPF9803**  
Cannot allocate object &2 in library &3.
- CPF9810**  
Library &1 not found.
- 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.
- CPF9871**  
Error occurred while processing.

Top



---

## Dsp Recovery for Access Paths (DSPRCYAP)

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

Parameters  
Examples  
Error messages

The Display Recovery for Access Paths (DSPRCYAP) command is used to show or print the access path recovery status information and target access path recovery time for the system and for all auxiliary storage pools (ASPs) that are currently active or have available status on the system. Additionally, the output will include up to 500 access paths with the largest estimated access path recovery time which are not eligible for system-managed access-path protection and why they are not eligible. Also, the output will include up to 500 access paths with the largest estimated access path recovery time which are currently being protected by system-managed access-path protection.

The system uses no more than the specified target access path recovery time when recovering access paths during an initial program load (IPL) or vary on of an independent ASP after an abnormal system end. Because access path recovery time is a target, performance may range around the target.

The time taken to rebuild access paths exposed while running the Copy File (CPYF), the Reorganize Physical File Member (RGZPFM), or the Restore Object (RSTOBJ) commands is not considered in the target access path recovery time of access paths protected with this command.

For more information on using this command, see the "Journal management" article in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

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

Top

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *_PRINT _	Optional, Positional 1
ASP	ASP device	Character value, *ALL, *ASPGRP	Optional
ASPGRP	ASP group	Name	Optional

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting work station or printed with the job's spooled output.

\*     The output requested by an interactive job is shown on the display. The output requested by a  
\_     batch job is printed with the job's spooled output.

#### \*PRINT

The output is printed with the job's spooled output. File QSYSPRT in QSYS is used for printed output.

Top

---

## ASP device (ASP)

Specifies the ASP containing the access paths to be displayed.

**Note:** This parameter is valid only if OUTPUT(\*PRINT) is selected.

**\*ALL** The output requested will be the top 500 access paths across all active or available ASPs.

**\*ASPGRP**

The output requested will be the top 500 access paths across the input ASP group.

**ASP-identifier**

Specify a value ranging from 1 through 32 to specify the identifier of the basic user ASP from which to list the top 500 access paths. Or specify the ASP device name of the independent user ASP from which to list the top 500 access paths. Valid values depend on the ASPs active or available on the system.

**Note:** The value 1 is the system ASP. Values 2 to 32 denote all basic user ASPs. Independent ASPs are entered as device names.

Top

---

## ASP group (ASPGRP)

Specifies the ASP Group containing the access paths to be displayed.

**Note:** This parameter is valid only if OUTPUT(\*PRINT) and ASP(\*ASPGRP) are specified.

**ASP-group-name**

Specifies the name of the auxiliary storage pool(ASP) group from which to list the top 500 access paths. The ASP group name is the name of the primary independent ASP device within the ASP group.

Top

---

## Examples

### Example 1: Displaying All Recovery Times for Access Paths

```
DSPRCYAP
```

This command shows the target access path recovery times and recovery status information for the system and active or available auxiliary storage pools. Output from the command is shown on the workstation if the command is run interactively, or printed with the job's spooled output if the command is run in batch.

### Example 2: Printing Recovery Times for Access Paths for an ASP Group

```
DSPRCYAP OUTPUT(*PRINT) ASP(*ASPGRP) ASPGRP(WAREHUS1)
```

This command shows the target access path recovery times and recovery status information for the independent ASP group WAREHUS1. Output from the command is printed with the job's spooled output.

Top



---

## Error messages

### \*ESCAPE Messages

#### **CPF70FB**

No authority to use command.

#### **CPF70F4**

Error occurred.

#### **CPF700F**

Access path recovery time for &1 set to \*NONE.

#### **CPF702E**

Access path recovery times set to system defaults.

#### **CPF9814**

Device &1 not found.

#### **CPF9825**

Not authorized to device &1.

#### **CPF9871**

Error occurred while processing.

#### **CPF88ED**

Device description &1 not correct for operation.

Top



---

## Display RDB Directory Entries (DSPRDBDIRE)

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

Parameters  
Examples  
Error messages

The Display Relational Database Directory Entry (DSPRDBDIRE) command allows you to display one entry, generic entries, or all entries in the relational database (RDB) directory.

Top

---

### Parameters

Keyword	Description	Choices	Notes
RDB	Entry	Generic name, name, <u>*ALL</u>	Optional, Positional 1
OUTPUT	Output	*, *PRINT, *OUTFILE	Optional, Positional 2
OUTFILE	File to receive output	Qualified object name	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
OUTMBR	Output member options	Element list	Optional
	Element 1: Member to receive output	Name, <u>*FIRST</u>	
	Element 2: Replace or add records	<u>*REPLACE</u> , *ADD	

Top

---

### Entry (RDB)

Specifies the relational database directory entry. If an entry has no alias, the relational database name is used as the entry name. If an entry has an alias, the alias is used as the entry name.

This is a required parameter.

The possible values are:

\*ALL All entries in the relational database directory.

*generic\*-relational-database-name*

Specify the generic name of the relational database entries. A generic name is a character string that contains one or more characters followed by an asterisk (\*).

*relational-database-name*

Specify the name of the relational database directory entry.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

The possible values are:

\*  
\_ The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

### \*PRINT

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

### \*OUTFILE

The output is directed to an output file.

Top

---

## File to receive output (OUTFILE)

Specifies the name and library of the output file to which the output of this command is directed. If the file does not exist, it is created. If a new file is created, system file QADSPDE in system library QSYS with a record format name of RWRDDSP is used as a model. If the file already exists, it must have this format.

This parameter is valid only if you specify \*OUTFILE on the **Output** prompt (OUTPUT parameter).

The possible library values are:

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

### \*CURLIB

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

### *library-name*

Specify the name of the library in which the output file is to be located.

### *database-file-name*

Specify the name of the file in which the output is to be located.

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.

The possible **member to receive output** values are:

### \*FIRST

The first member of the specified file is used.

### *member-name*

Specify a maximum of 10 characters for the name of the output file member in which the requested information is to be located.

The possible **replace or add records** values are:

**\*REPLACE**

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

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

Top

---

## Examples

### Example 1: Directing Information to an Output File

```
DSPRDBDIRE OUTPUT(*PRINT)
```

This command directs information from all of the relational database directory entries to a spooled file.

### Example 2: Directing Information to an Output File

```
DSPRDBDIRE OUTPUT(*OUTFILE) OUTFILE(SAVEDIR)
```

This command directs all of the relational database directory entries to an output file named SAVEDIR. This is the usual method for backing up the contents of the relational database directory. The entries can be restored using a CL program that reads the information from the output file and issues Add Relational Database Directory Entry (ADDRDBDIRE) commands to add the information back into the relational database directory.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF3EC3

Display relational database directory entry failed.

Top



---

## Display Remote Definition (DSPRMTDFN)

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

Parameters  
Examples  
Error messages

The Display Remote Definition (DSPRMTDFN) command allows the user to display or print remote definitions for a system. The output can be displayed, printed, or directed to a database file.

Top

---

### Parameters

Keyword	Description	Choices	Notes
SYSTEM	System name	<i>Element list</i>	Required, Positional 1
	Element 1: System name	<i>Character value, *ANY, *ALL</i>	
	Element 2: System group	<i>Character value</i>	
OUTPUT	Output	<i>*, *PRINT, *OUTFILE</i>	Optional, Positional 2
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>*ADD, *REPLACE</i>	

Top

---

### System name (SYSTEM)

Specifies the system name and system group of the remote system being displayed.

The possible values are:

**\*ANY** Displays the default definition for a remote system not covered by the other entries.

**\*ALL** Displays all definitions for remote systems.

The possible **System Name** value is:

*system-name*

Specify the name of the remote system to be displayed.

The possible **System Group** value is:

*system-group*

Specify the group name of the remote system to be displayed. Do not specify this value if the group name is blank.

---

## Output (OUTPUT)

Specifies whether the output from this command is displayed, printed, or directed to a database file. More information on this parameter is in "Appendix A, Expanded Parameter Descriptions" in the CL Reference.

The possible values are:

\*  
- Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

### \*PRINT

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

### \*OUTFILE

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

---

## File to receive output (OUTFILE)

Specifies the qualified name of the database file to which the output of this command is directed. If the file does not exist, the system creates a file in the specified library. If a new file is created, system file QAOCRMTDFN in system library QSYS with a record format name of RMTDFN is used as a model. If the file already exists, it must have this format.

The possible library values are:

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

### \*CURLIB

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

### *library-name*

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

The possible value is:

### *file-name*

Specify the name of the output file.

---

## Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed.

The possible **member to receive output** values are:

### \*FIRST

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



### *member-name*

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

The possible **operation to perform on member** values are:

#### \*REPLACE

The output data replaces existing records in the specified member.

**\*ADD** The output data is added after existing records in the specified member.

Top

---

## Examples

### Example 1: Displaying a Specific Remote Definition

```
DSPRMTDFN  SYSTEM(RCHAS1)
```

This command displays the current attributes for remote system RCHAS1.

### Example 2: Writing a Definition to an Output File

```
DSPRMTDFN  SYSTEM(*ALL)  OUTPUT(*OUTFILE)  OUTFILE(RMTDFNOUT)
```

This command writes the current attributes for all defined remote systems to the output file RMTDFNOUT.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF6A50

Error was found during display file or printer file operation.

#### CPF6DCA

SYSTEM parameter cannot be local system.

#### CPF6DCC

Remote definition for system &1 &2 not found.

#### CPF9860

Error occurred during output file processing.

#### CPF9899

Error occurred during processing of command.

Top



---

## Display S/36 Configuration (DSPS36)

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

Parameters  
Examples  
Error messages

The Display System/36 (DSPS36) command allows the user to show or print the description of the System/36 environment configuration. The description includes System/36 printers, display stations, general environment values, and (if the user is authorized) MRT security values.

[Top](#)

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *PRINT _	Optional, Positional 1

[Top](#)

---

### Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting work station or printed with the job's spooled output.

**\*:** The output is displayed at the requesting work station if requested by an interactive job. If this is not an interactive job, the output is printed with the job's spooled output.

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

[Top](#)

---

### Examples

DSPS36 OUTPUT(\*)

This command allows the user in an interactive job to display the System/36 environment description.

[Top](#)

---

### Error messages

None

[Top](#)



---

## Display Save File (DSPSAVF)

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

Parameters  
Examples  
Error messages

The Display Save File (DSPSAVF) command displays the save information in a save file. This includes summary information about the save operation and a description of each object saved to the save file.

### Restrictions:

- You must have use (\*USE) authority for the save file and read (\*READ) authority for the specified library.

Top

---

## Parameters

Keyword	Description	Choices	Notes
FILE	Save file	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Save file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTPUT	Output	<i>*, *PRINT _</i>	Optional, Positional 2

Top

---

## Save file (FILE)

Specifies the save file to be displayed.

This is a required parameter.

### Qualifier 1: Save file

*name* Specify the name of the save 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 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

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_ The output is displayed for interactive jobs or printed with the job's spooled output for non-interactive jobs.

**\*PRINT**

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

Top

---

## Examples

```
DSPSAVF FILE(ONLINE) OUTPUT(*PRINT)
```

This command shows the objects saved to save file ONLINE. The output is printed with the job's spooled output.

Top

---

## Error messages

### \*ESCAPE Messages

**CPD375A**

Media error on save media.

**CPF3704**

Request ended; data management error occurred.

**CPF3743**

File cannot be restored, displayed, or listed.

**CPF3782**

File &1 in &2 not a save file.

**CPF3782**

File &1 in &2 not a save file.

**CPF3792**

Information not displayed. Error occurred.

**CPF3793**

Machine or ASP storage limit reached.

**CPF3812**

Save file &1 in &2 in use.

**CPF9806**

Cannot perform function for object &2 in library &3.

**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.

**CPF9850**

Override of printer file &1 not allowed.

**CPF9851**

Overflow value for file &1 in &2 too small.

Top





---

## Display Subsystem Description (DSPSBSD)

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

Parameters  
Examples  
Error messages

The Display Subsystem Description (DSPSBSD) command displays the information contained in a subsystem description. The types of information (which are shown on separate displays) include: operational attributes, pool definitions, autostart job entries, work station entries (by name and type), job queue entries, routing entries, communications entries, remote location entries, and prestart job entries. If this command is entered in a batch job, **all** available information is printed with the job's spooled output.

### Restrictions:

1. To use this command, you must have:
  - object operational (\*OBJOPR) and read (\*READ) authority to the specified subsystem description and execute (\*EXECUTE) authority to the library containing that subsystem description.

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>	
OUTPUT	Output	<i>*, *PRINT _</i>	Optional, Positional 2

Top

---

## Subsystem description (SBSD)

Specifies the name and library of the subsystem description being displayed.

This is a required parameter.

### Qualifier 1: Subsystem description

*name* Specify the name of the subsystem description.

### 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 subsystem description. If no library is specified as the current library for the thread, library QGPL is used.

*name* Specify the library where the subsystem description is located.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_ The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

### \*PRINT

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

Top

---

## Examples

```
DSPSBSD  SBSDB(LIB6/ORDER)  OUTPUT(*)
```

This command (if entered from a batch job) sends a complete set of display information about the subsystem description named ORDER (stored in LIB6 library) to the job's spooling queue for printing. The information includes the subsystem's attributes, all of the job entries, and all of the routing entries currently in the subsystem description. If the command is entered in an interactive job, the subsystem description menu is shown on a display from which an option may be chosen.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1619

Subsystem description &1 in library &2 damaged.

#### CPF1692

Subsystem description &1 not displayed.

#### CPF9850

Override of printer file &1 not allowed.

#### CPF9851

Overflow value for file &1 in &2 too small.

#### CPF9861

Output file &1 created in library &2.

#### CPF9871

Error occurred while processing.

Top

---

# Display Security Attributes (DSPSECA)

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

Parameters  
Examples  
Error messages

The Display Security Attributes (DSPSECA) command displays security attributes of the system such as:

- The user ID number that will be used the next time a user ID number is generated for a user profile.
- The group ID number that will be used the next time a group ID number is generated for a user profile.
- The security level of the machine. The pending security level is shown if it is different than the current security level.
- The password level of the machine. The pending password level is shown if it is different than the current password level.
- The indicator for whether or not security related system values can be changed.
- The indicator for whether or not digital certificates can be added to a certificate store and whether or not the password for a certificate store can be reset using Digital Certificate Manager (DCM).
- The indicator for whether or not a service tools user ID with a default password that is expired can change its own password.

Top

---

## Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *PRINT _	Optional, Positional 1

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_     The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

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

Top

---

## Examples

DSPSECA   OUTPUT(\*PRINT)

This command prints the current security attributes.

Top

---

## Error messages

None

[Top](#)

---

# Display Security Auditing (DSPSECAUD)

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

Parameters  
Examples  
Error messages

The Display Security Auditing (DSPSECAUD) command displays current information about the security audit journal and the current settings for the system values that control what is being audited on the system.

**Restriction:** You must have audit (\*AUDIT) special authority to run this command.

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *_PRINT	Optional, Positional 1

[Top](#)

---

## Output (OUTPUT)

Specifies where the output from the command is sent.

\*        The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).  
\_

**\*PRINT**

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

[Top](#)

---

## Examples

DSPSECAUD

This command displays current information about the security audit journal and the current settings for the system values that control what is being audited on the system.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPFB304

User does not have required special authorities.

[Top](#)



---

## Display Software Resources (DSPSFWRSC)

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

Parameters  
Examples  
Error messages

The Display Software Resources (DSPSFWRSC) command allows you to show, print, or write to an output file the list of installed software resources.

Top

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *PRINT, *OUTFILE	Optional, Positional 1
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional, Positional 2
	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

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

The possible values are:

\*     The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).  
-     

#### \*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:** If OUTPUT(\*OUTFILE) is used, the name of the database file is required.

Top

---

## File to receive output (OUTFILE)

Specifies the name and library of 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 a new file is created, the system uses the physical file QARZLCOF in the library list as a model file. The file has a record format name of QARZLCGD. Field level information can be obtained using the Display File Field Description (DSPFFD) command and specifying QARZLCOF as the file name and \*LIBL as the library. If this function creates the file, the text says "Output file for DSPSFWRSC". The authority for users other than those who have specific authority, group authority, or authority from an authorization list, is the same as the create authority specified for the library in which the file is created. This parameter is valid only if OUTPUT(\*OUTFILE) is specified.

The name of the database 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.

Top

---

## Output member options (OUTMBR)

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

The possible values are:

**\*FIRST**

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

*member-name*

Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it. If the member exists, you have the option to add records to the end of the existing member, or clear the existing member and then add the new records.

The optional values are:

**\*REPLACE**

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

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

Top

---

## Examples

```
DSPSFWRSC OUTPUT(*OUTFILE) OUTFILE(*CURLIB/NAMES)
```

This command sends the output from the command to the first member of the file NAMES in the current library. The output replaces the information in the member.



---

## Error messages

### \*ESCAPE Messages

#### CPF0C4A

Product record not found.

#### CPF0C4B

Product availability object &2/&1 recovery required.

#### CPF0C4C

Cannot allocate object &1 in library &2.

#### CPF0C4D

Error occurred while processing object &1 in library &2.

#### CPF0C54

Data in product record not correct.

#### CPF9860

Error occurred during output file processing.

#### CPF9871

Error occurred while processing.



---

# Display Sphere of Control Sts (DSPSOCSTS)

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

Parameters  
Examples  
Error messages

The Display Sphere of Control Status (DSPSOCSTS) command shows the status of the sphere of control including primary, default, back up, and requested nodes.

More information about displaying the sphere of control status and using the DSPSOCSTS command is in the Alerts Support book, SC41-5413.

Top

---

## Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *_PRINT	Optional, Positional 1
DETAIL	Detail	* <u>BASIC</u> , *FULL	Optional

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_ Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

### \*PRINT

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

Top

---

## Detail (DETAIL)

Specifies whether the output from the command is printed as a list of nodes or as a detailed description of each node in the sphere of control. This parameter is valid only when OUTPUT(\*PRINT) is specified.

### \*BASIC

A list of nodes in the sphere of control is printed.

\*FULL A list of nodes in the sphere of control with detailed information for each node is printed.

Top

---

## Examples

DSPSOCSTS

This command shows the Sphere of Control Status display at the requesting work station.

[Top](#)

---

## Error messages

None

[Top](#)

---

## Display Spooled File (DSPSPLF)

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

Parameters  
Examples  
Error messages

The Display Spooled File (DSPSPLF) command shows the data records in the specified spooled file. The current contents of the file (data records) can be displayed any time an entry for the spooled file is on the output queue. The screen provides various functions to display various parts of the file and to scan for a specific character string.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FILE	Spooled file	<i>Name</i>	Required, Positional 1
JOB	Job name	Single values: * Other values: <i>Qualified job name</i>	Optional, Positional 2
	Qualifier 1: Job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
SPLNBR	Spooled file number	1-999999, * <u>ONLY</u> , *LAST, *ANY	Optional, Positional 3
JOBSYSNAME	Job system name	<i>Name</i> , * <u>ONLY</u> , *CURRENT, *ANY	Optional
CRTDATE	Spooled file created	Single values: * <u>ONLY</u> , *LAST Other values: <i>Element list</i>	Optional
	Element 1: Creation date	<i>Date</i>	
	Element 2: Creation time	<i>Time</i> , * <u>ONLY</u> , *LAST	
FOLD	Fold records	* <u>NO</u> , *YES	Optional

Top

---

### Spooled file (FILE)

Specifies the spooled file which has its records displayed.

This is a required parameter.

*name* Specify the file name of the spooled file to be displayed.

Top

---

### Job name (JOB)

Specifies the name of the job that created the spooled file whose data records are to be displayed.

#### Single values

\*  
\_ The job that issued this command is the job that created the spooled file.

#### **Qualifier 1: Job name**

*name* Specify the name of the job that created the spooled file.

#### **Qualifier 2: User**

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

#### **Qualifier 3: Number**

**000000-999999**

Specify the system-assigned job number.

Top

---

## **Spooled file number (SPLNBR)**

Specifies the number of the job's spooled file that is to be displayed.

#### **\*ONLY**

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

#### **\*LAST**

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

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

**1-999999**

Specify the number of the spooled file having the specified file name whose data records are displayed.

Top

---

## **Job system name (JOBSYSNAME)**

Specifies the system where the job that created the spooled file (JOB parameter) ran. This parameter is considered after the job name, user name, job number, spooled file name, and spooled file number parameter requirements have been met.

#### **\*ONLY**

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

#### **\*CURRENT**

The spooled file created on the current system with the specified job name, user name, job number, spooled file name, spooled file number, and create date and time is used.

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

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

Top

---

## Spooled file created (CRTDATE)

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

### Single values

#### \*ONLY

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

#### \*LAST

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

### Element 1: Creation date

*date* Specify the date the spooled file was created.

### Element 2: Creation time

#### \*ONLY

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

#### \*LAST

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

*time* Specify the time the spooled file was created.

Top

---

## Fold records (FOLD)

Specifies whether the first display has the records folded (wrapped) if they are longer than the length of the display line.

\*NO The records are not folded. When the length of the record is longer than one line, the remaining positions of the record are not shown.

\*YES The initial display shows the first record folded on one or more display lines if it is longer than one line.

Top

---

## Examples

```
DSPSPFLF FILE(QPRINT) JOB(PAYROLL01) SPLNBR(4) FOLD(*NO)
```

In this example, the spooled file QPRINT is displayed. The file is the fourth file produced by the job PAYROLL01. The record positions that are longer than the length of the display line are truncated on the first display.

Top

---

## Error messages

### \*ESCAPE Messages

**CPF2207**

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

**CPF3303**

File &1 not found in job &5/&4/&3.

**CPF3308**

Error occurred when trying to display data.

**CPF3309**

No files named &1 are active.

**CPF3330**

Necessary resource not available.

**CPF3340**

More than one file with specified name found in job &5/&4/&3.

**CPF3342**

Job &5/&4/&3 not found.

**CPF3343**

Duplicate job names found.

**CPF3344**

File &1 number &8 no longer in the system.

**CPF3359**

Not able to display data.

**CPF3386**

File &1 in &2 not a data base file.

**CPF3387**

Cannot display data in file &1 in &2.

**CPF3394**

Cannot convert spooled file data.

**CPF33F9**

Error occurred while displaying file &1 number &6.

**CPF3427**

Job &5/&4/&3 not interactive job.

**CPF3428**

DSPSPLF command ended for file &1 number &8.

**CPF3429**

File &1 number &7 cannot be displayed, copied, or sent.

**CPF3434**

Data in file &1 in &2 member &3 not in required format.

**CPF3435**

Requested data not found in file &1 in &2 member &3.

**CPF3478**

File &1 not found in job &5/&4/&3 on output queue &6 in library &7.

**CPF3492**

Not authorized to spooled file.



**CPF7D41**

Error occurred while logging order assistance request.

**CPF7D42**

Error occurred while performing database operation.

**CPF9812**

File &1 in library &2 not found.

**CPF9815**

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

**CPF9845**

Error occurred while opening file &1.

**CPF9846**

Error while processing file &1 in library &2.

**CPF9871**

Error occurred while processing.

Top



---

# Display Service Attributes (DSPSRVA)

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

Parameters  
Examples  
Error messages

The Display Service Attributes (DSPSRVA) command displays information about how the system is set up to:

- The connection number to report to external support
- The connection number to call back this system
- Whether problem analysis routines should run automatically when a failure occurs
- How the specified service provider should be notified of problems
- The connection number to the service provider
- When PTFs should be installed
- Where critical system messages are sent
- Whether PTF save files and cover letters should be copied into \*SERVICE when PTFs are loaded from a tape or optical device

Top

---

## Parameters

None

Top

---

## Examples

DSPSRVA

This command displays the current service attributes for the system.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF7D41

Error occurred while logging order assistance request.

#### CPF7D42

Error occurred while performing database operation.

#### CPF9871

Error occurred while processing.

Top



---

## Display Service Agent (DSPSRVAGT)

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

Parameters  
Examples  
Error messages

The Display Service Agent (DSPSRVAGT) command allows a user to display several aspects of Service Agent. The area to be displayed is specified by the **Type (TYPE)** parameter.

Top

---

### Parameters

Keyword	Description	Choices	Notes
TYPE	Type	*DEVICE, *SRVIN, *MAINT, *PRDACTLOG, *SRVREGINF, *PRBRPT	Required, Positional 1
DEVICE	Device	Character value	Optional
CATEGORY	Category	*DASD, *TAPE, *PROCESSOR, *OPTICAL, *FSIOP, *OTHER	Optional
SENSEFMT	Sense byte format	0, 4, 2, 8, C	Optional
SYSNAME	System or logical partition	Character value, <u>*LOCAL</u>	Optional
OUTPUT	Output	*, *PRINT	Optional
SRVREP	IBM Service representative	Character value, <u>*BLANK</u>	Optional
SRVTELNR	IBM Service telephone number	Character value, <u>*BLANK</u>	Optional
RECOMMEND	Recommendations	Character value, <u>*BLANK</u>	Optional
ERRLOGID	Error log identifier	Character value	Optional

Top

---

### Type (TYPE)

Specifies the aspect of Service Agent to be displayed.

This is a required parameter.

#### \*DEVICE

Information from the Service Agent threshold table for a device is to be displayed.

#### \*SRVIN

Information about Service Agent service information collection and transmission is displayed.

#### \*MAINT

The IBM Service preventive maintenance checklist is to be displayed or printed. This is a list of actions suggested during a preventive maintenance call on the system or logical partition. This list should be printed and given to the customer after each preventive maintenance service call.

#### \*PRDACTLOG

A record from the Product Activity Log is to be displayed or printed.

#### \*SRVREGINF

Information needed to register a system or logical partition as a Service Agent is to be displayed.

---

## Device (DEVICE)

Specifies the device for which the threshold table information is to be displayed.

**Note:** This is a required parameter when TYPE(\*DEVICE) is specified.

### *character-value*

Specify the four-character device type for which the threshold table information is to be displayed. For example, DEVICE(2420) might be specified for a 2420 tape device.

---

## Category (CATEGORY)

Specifies the type of device for which the threshold table information is to be displayed.

**Note:** This is a required parameter when TYPE(\*DEVICE) is specified.

### \*DASD

The device is a DASD device.

\*TAPE The device is a tape device.

### \*PROCESSOR

The device is a processor.

### \*OPTICAL

The device is an optical device.

### \*FSIOP

The device is an FSIOP device.

### \*OTHER

The device is other than one of the above listed devices.

---

## Sense byte format (SENSEFMT)

Specifies the format of the volume statistical data for tape devices.

**Note:** This parameter is valid only when CATEGORY(\*TAPE) is specified.

0 The device does not report removable media statistics.

4 The format is for a 1/4" cartridge tape device.

2 The format is for a 1/2" reel tape device.

8 The format is for an 8 mm tape device.

C The format is for a 1/2" cartridge tape device.

---

## System or logical partition (SYSNAME)

Specifies the name of the system or logical partition for which information is to be displayed.

**Note:** This parameter is valid only when TYPE(\*SRVINFL) is specified.

### \*LOCAL

Information for the local system or logical partition is to be displayed.

### *character-value*

Specify the name of the system or logical partition for which information is to be displayed.

Top

---

## Output (OUTPUT)

Specifies where the output from the command is to be directed.

**Note:** This parameter is valid only when TYPE(\*MAINT) or TYPE(\*PRDACT) is specified.

\*  
\_ The output is displayed for interactive jobs or printed with the job's spooled output for non-interactive jobs.

### **\*PRINT**

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

Top

---

## IBM Service representative (SRVREP)

Specifies the name of the IBM service representative making this service call or the name of the IBM service representative the customer should contact when they have questions concerning the service performed.

**Note:** This parameter is valid only when TYPE(\*MAINT) and OUTPUT(\*PRINT) are specified.

### \*BLANK

No value is specified.

### *character-value*

Specify the name of the IBM Service contact. This value will be included in the output when OUTPUT(\*PRINT) is specified.

Top

---

## IBM Service telephone number (SRVTELNBR)

Specifies the telephone number the customer should call to contact IBM Service.

**Note:** This parameter is valid only when TYPE(\*MAINT) and OUTPUT(\*PRINT) are specified.

### \*BLANK

No value is specified.

### *character-value*

Specify the complete telephone number sequence to contact IBM Service. This value will be included in the output when OUTPUT(\*PRINT) is specified.

---

## Recommendations (RECOMMEND)

Specifies any recommendations the IBM Service representative has for the customer.

**Note:** This parameter is valid only when TYPE(\*MAINT) and OUTPUT(\*PRINT) are specified.

### \*BLANK

No value is specified.

### *character-value*

Specify any recommendations IBM Service wishes to make to the customer. This value will be formatted and included in the output when OUTPUT(\*PRINT) is specified.

Top

---

## Error log identifier (ERRLOGID)

Specifies the record identifier for the Product Activity Log record that is to be displayed.

**Note:** This parameter is required when TYPE(\*PRDACTLOG) is specified.

### *character-value*

Specify the identifier of the Product Activity Log record to be displayed. Identifiers may be found using the Work with Service Agent (WRKSRVAGT) command with TYPE(\*EVENT) specified, or by using the Start Service Tools (STRSST) command.

Top

---

## Examples

```
DSPSRVAGT TYPE(*DEVICE) DEVICE(2420) CATEGORY(*TAPE)
```

This command will display the information in the Service Agent threshold table for device 2420, a tape device.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9899

Error occurred during processing of command.

Top



---

## Display Service Program (DSPSRVPGM)

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

Parameters  
Examples  
Error messages

The Display Service Program (DSPSRVPGM) command displays information about a service program, including the creation and processing attributes of the service program, information about the compiler, and the size of the service program.

### Restrictions:

- You must have execute (\*EXECUTE) authority to the service program being displayed, or use (\*USE) authority when DETAIL(\*MODULE) is specified.
- You must have read (\*READ) authority to the library in which the service program exists.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SRVPGM	Service program	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Service program	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTPUT	Output	<i>*, *PRINT</i> <i>_, *PRINT</i>	Optional, Positional 2
DETAIL	Detail	Single values: <b>*ALL</b> Other values (up to 10 repetitions): *BASIC, *SIZE, *MODULE, *SRVPGM, *PROCEXP, *DTAEXP, *ACTGRPEXP, *ACTGRPIMP, *SIGNATURE, *COPYRIGHT	Optional

Top

---

## Service program (SRVPGM)

Specifies the service program for which information is displayed or printed.

This is a required parameter.

### Qualifier 1: Service program

*name* Specify the name of the service program.

### Qualifier 2: Library

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

### \*CURLIB

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

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

---

## Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting work station or printed with the job's spooled output.

This is a required parameter.

\*  
- Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

### \*PRINT

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

---

## Detail (DETAIL)

Specifies the type of information displayed for the service program.

This is a required parameter.

### Single values

\*ALL All of the DETAIL information types (\*BASIC, \*SIZE, \*MODULE, \*SRVPGM, \*PROCEXP, \*DTAEXP, \*ACTGRPEXP, \*ACTGRPIMP, \*SIGNATURE, and \*COPYRIGHT) are shown on the display. If you chose to have the information displayed on a screen, you would be able to scroll through the information for each type of information, but would have to press Enter (or PF12) to go from one information section to the next.

### Other values (up to 8 repetitions)

#### \*BASIC

General service program information is shown.

\*SIZE The size and size limits for this service program are shown.

#### \*MODULE

A list of the module objects bound by this service program is shown. The library shown for each module is the library that the module was in when the service program was first created. If the module has been replaced by a module from a different library, this library name remains the name of the library that the module was in when the service program was created.

#### \*SRVPGM

A list of the service program objects bound by this service program is shown.

#### \*PROCEXP

A list of the procedures exported from a service program and specified in the binding language is shown. The exports are only for the current signature.

#### \*DTAEXP

A list of the data items exported from a service program and specified in the binding language is shown. The exports are only for the current signature.

#### \*ACTGRPEXP

A list is shown of the data items exported to the activation group specified in the data export entry in the binding specifications.

#### \*ACTGRPIMP

A list is shown of the imports that are resolved by weak exports that had been exported to the activation group directory.

#### \*SIGNATURE

A list of the signatures for this service program is shown. The first signature in the list is the current signature.

#### \*COPYRIGHT

A list of the copyrights for this service program is shown.

Top

---

## Examples

DSPSRVPGM SRVPGM(COACH)

This command displays a service program object named COACH.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2150

Object information function failed.

#### CPF2151

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

#### CPF8122

&8 damage on library &4.

#### CPF8123

Damage on object information for library &4.

#### CPF813D

Service program &4 in &9 damaged.

#### CPF9801

Object &2 in library &3 not found.

#### CPF9802

Not authorized to object &2 in &3.

#### CPF9803

Cannot allocate object &2 in library &3.

#### 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.

#### CPF9810

Library &1 not found.

#### CPF9820

Not authorized to use library &1.

**CPF9830**

Cannot assign library &1.

**CPF9871**

Error occurred while processing.

[Top](#)

---

## Display Service Status (DSPSRVSTS)

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

Parameters  
Examples  
Error messages

The Display Service Status (DSPSRVSTS) command shows information about the current service status of the specified job. This includes the name of the job it is servicing or the name of the job servicing the specified job.

### Restrictions:

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

Top

---

## Parameters

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

Top

---

## Job name (JOB)

Specifies which job is to have its service status shown.

### Single values

\*        Status information is shown about the job in which the command is entered.

### Qualifier 1: Job name

*name*    Specify the job name of the job to be shown. If no user name or job number are specified, all of the jobs currently in the system are searched for the simple job name; the specified job name must be unique within the system.

### Qualifier 2: User

*name*    Specify the user name of the job to be shown.

### Qualifier 3: Number

**000000-999999**

Specify the job number of the job to be shown.

Top

---

## Duplicate job option (DUPJOB OPT)

Specifies the action taken when duplicate jobs are found by this command.

### **\*SELECT**

The selection display is shown when duplicate jobs are found during an interactive session. Otherwise, an escape message is issued.

**\*MSG** An escape message is issued when duplicate jobs are found.

Top

---

## Examples

DSPSRVSTS

This command shows the service status information for the job from which the command is entered.

Top

---

## Error messages

### **\*ESCAPE Messages**

#### **CPF3520**

Job not found.

#### **CPF3524**

More than one job with specified name found.

#### **CPF3925**

Cannot open file &1.

#### **CPF3950**

Error message &2 received for file &1. Request ended.

Top

---

## Display Server Auth Entries (DSPSVRAUTE)

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

Parameters  
Examples  
Error messages

The Display Server Authentication Entries (DSPSVRAUTE) command shows a list of server authentication entries for a specified user profile.

Top

---

### Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	<i>Simple name</i> , *CURRENT	Optional, Positional 1
OUTPUT	Output	*, *_PRINT	Optional

Top

---

### User profile (USRPRF)

Specifies the user profile for which the server authentication entries are to be displayed.

#### \*CURRENT

The server authentication entries for the user running this command are to be displayed.

*name* Specify the name of the user whose server authentication entries are to be displayed.

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_ The output is displayed for interactive jobs or printed with the job's spooled output for non-interactive jobs.

#### \*PRINT

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

Top

---

### Examples

DSPSVRAUTE

This command will show the server authentication entries for the currently running user.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFA0AA

Error occurred while attempting to obtain space.

#### CPF22F0

Unexpected errors occurred during processing.

#### CPF2204

User profile &1 not found.

#### CPF2213

Not able to allocate user profile &1.

#### CPF2217

Not authorized to user profile &1.

[Top](#)



---

## Display System Status (DSPSYSSTS)

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

Parameters  
Examples  
Error messages

The Display System Status (DSPSYSSTS) command allows the user to display or print information about the current status of the system.

Top

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	<u>*</u> , *PRINT	Optional, Positional 1
RESET	Reset status statistics	* <u>N</u> O, *YES	Optional
ASTLVL	Assistance level	* <u>P</u> RV, *USRPRE, *BASIC, *INTERMED, *ADVANCED	Optional

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*     The output is displayed for interactive jobs or printed with the job's spooled output for  
-     non-interactive jobs.

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

Top

---

### Reset status statistics (RESET)

Specifies whether system status statistics fields are reset to zero, as if this is the first occurrence of the DSPSYSSTS command in this job. The value specified for this parameter does not affect the information presented for \*BASIC assistance level.

\*NO     The system status statistics are not reset.

\*YES     The system status statistics are reset. This will also reset the status statistics on the Work with System Status (WRKSYSSTS) command and the QWCRSSTS API.

Top

---

### Assistance level (ASTLVL)

Specifies which user interface to use.

\*PRV     The previously used assistance level is used.

**\*USRPRF**

The assistance level defined in the user profile is used.

**\*BASIC**

The Operational Assistant user interface is used.

**\*INTERMED**

The system user interface is used.

**\*ADVANCED**

The system user interface expert mode is used.

[Top](#)

---

## Examples

DSPSYSSTS OUTPUT(\*PRINT)

This command prints the current system status information.

[Top](#)

---

## Error messages

None

[Top](#)

# Display System Value (DSPSYSVAL)

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

Parameters  
 Examples  
 Error messages

The Display System Value (DSPSYSVAL) command displays the name and the value of the specified system value.

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, QDSPSGNIN, QDYNPTYADJ, QDYNPTYSCD, QENDJOBVMT, QFRCCVNRST, QHOUR, QHSTLOGSIZ, QIGC, QIGCCDEFNT, QIGCFNTSIZ, QINACTMSGQ, QINACTITV, QIPLDATTIM, QIPLSTS, QIPLTYPE, QJOBMSGQFL, QJOBMSGQMX, QJOBMSGQSZ, QJOBMSGQTL, QJOBSPLA, QKBDBUF, 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, QPRTXT, QPWDEXPTV, QPWDLMTAJC, QPWDLMTCHR, QPWDLMTREP, QPVDLVL, QPWDMAXLEN, QPWDMINLEN, QPWDPOSDIF, QPWDRQDDGT, QPWDRQDDIF, QPWVLDLPGM, QPWRDWNLMT, QPWRRSTIPL, QQRYDEGREE, QQRYTIMLMT, QRCLSPLSTG, QRETSVRSEC, QRMTIPL, QRMTSIGN, QRMTSRVATR, QSAVACCPH, QSCANFS, QSCANFCTL, QSCPFCONS, QSECOND, QSECURITY, QSETJOBATR, QSFWERRLOG, QSHRMEMCTL, QSPCENV, QSPLFACN, QSRLNBR, QSRTSEQ, QSRVDM, QSTGLOWACN, QSTGLOWLMT, QSTRPRTWTR, QSTRUPPGM, QSTSMSG, QSVRAUTITV, QSYSLIBL, QTHDRSCADJ, QTHDRSCAFN, QTIMADJ, QTIME, QTIMSEP, QTIMZON, QTOTJOB, QTSEPOOL, QUPSDLYTIM, QUPSMSGQ, QUSEADPAUT, QUSRLIBL, QUTCFFSET, QVYOBJRST, QYEAR	Required, Positional 1
OUTPUT	Output	*, *PRINT _,	Optional, Positional 2

---

## System value (SYSVAL)

Specifies the name of the system value that is displayed.

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 iSeries Information Center contains information on using job interrupt exit programs. The Change Job Interrupt Status (QWCCJITP) API in the iSeries Information Center 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.

## QALWOBJRST

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.
- \*JOBDA - Actions that affect a job are audited.
- \*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.
- \*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 iSeries Security Reference 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.
- \*JOBDDTA - Actions that affect a job are audited.
- \*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.
- \*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 iSeries Security Reference 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.

**QAUTOSPRPT**

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 QLUS 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 iSeries 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.

## QENDJOBMT

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 QENDJOBMT 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 and objects with validation errors will 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 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.

## QHOUR

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

## **QKBDBUF**

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. Changes made to this system value take effect immediately.

## **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 you can sign on at multiple devices.
- 1 means you cannot sign on at more than one device.

## **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, iSeries 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.

## QPRBHLDTIV

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. If enabled, more than one set of task data will be resident in each CPU. Some workloads may experience increased performance due to caching implications. **Note:** The operating system will set the system value to 0 on the next IPL if it detects that the hardware does not support multitasking. Setting the value to system controlled will allow the system to manage the multitasking. Changes made to this system value take effect at the next IPL.

- 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 iSeries Information Center at <http://www.ibm.com/eserver/series/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.

## 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.

## 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.

## 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.

## 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.

## 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 iSeries Security Reference 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. iSeries 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. iSeries 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.

#### **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.

#### **QPWDPOSDIF**

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.

#### **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.

#### **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.

#### **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 PWRDWNSYS \*IMMED. This is the time used to wait for power down to complete normally after either of the following happens:

- A Power Down System (PWRDWNSYS) command with \*IMMED specified for the **How to end** (OPTION) parameter is entered.
- A PWRDWNSYS 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 PWRDWNSYS 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, 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 iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

## QQRYPDEGREE

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 topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/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, 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 iSeries Information Center at <http://www.ibm.com/eserver/series/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.

## **QSCANFCTL**

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.

## 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.

## **QSVRAUTIV**

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. The system value QUTCOFFSET will be changed as well 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 at the next 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 powers only the PRC, IOP cards, and Load Source Disk.
- \*CALC means the appropriate wait time will be calculated.

- \*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 iSeries Information Center at <http://www.ibm.com/eserver/series/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

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*     The output is displayed for interactive jobs or printed with the job's spooled output for  
-     non-interactive jobs.

### \*PRINT

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

Top

---

## Examples

DSPSYSVAL   SYSVAL(QHOUR)

This command displays the current value of the system value QHOUR.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1028

&1 not valid for parameter SYSVAL.

#### CPF1074

SYSVAL(QMONTH) not valid for Julian date format.

#### 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.

#### CPF9871

Error occurred while processing.

Top

## Display Tape (DSPTAP)

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

Parameters  
 Examples  
 Error messages

The Display Tape (DSPTAP) command shows the volume label and data file label information that is contained on a standard labeled magnetic tape or the volume type and density. The information can be printed or shown on a display device. Additional information, including a description of each object saved to the tape file and summary information about the saved objects, can be displayed by specifying \*SAVRST on the **Data type** prompt (DATA parameter).

Top

### Parameters

Keyword	Description	Choices	Notes
DEV	Device	<i>Name</i>	Required, Positional 1
VOL	Volume identifier	<i>Character value, *MOUNTED</i>	Optional
LABEL	File label	<i>Character value, *ALL</i>	Optional
SEQNBR	Sequence number	Single values: *ALL Other values: <i>Element list</i>	Optional
	Element 1: Starting file sequence number	1-16777215, *FIRST	
	Element 2: Ending file sequence number	1-16777215, *ONLY, *LAST	
DATA	Data type	*LABELS, *SAVRST	Optional
OUTPUT	Output	*, *PRINT, *OUTFILE	Optional
ENDOPT	End of tape option	*REWIND, *UNLOAD	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 tape or media library device in which the volume being shown is placed.

This is a required parameter.

Top

---

## Volume identifier (VOL)

Specifies the tape volume to be displayed.

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

### \*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 category mounted by the Set Tape Category (SETTAPCGY) command. For a virtual tape device, the volume to be used is the currently mounted one, or if there is not a currently mounted volume, the next volume in loaded status in the image catalog will be used.

### *character-value*

Specify the volume identifier of the labeled volume. The volume identifier read from the tape is compared to this value. If the volume identifier specified is not found on the tape, an escape message is sent.

Top

---

## File label (LABEL)

Specifies the data file identifiers of the data files on the tape whose labels are shown. The data file identifier is stored in the label ahead of the data in the file.

**\*ALL** All data file identifiers on the tape specified on the **Tape device** prompt (DEV parameter) are shown.

### *character-value*

Specify the data file identifier (17 alphanumeric characters maximum) of the data file for which label information is shown.

Top

---

## Sequence number (SEQNBR)

Specifies, for volumes with multiple files, the range of sequence numbers of the data files on tape whose label information is displayed. If \*ALL is specified on the **File label** prompt (LABEL parameter), all labels for the files within the specified sequence number range are displayed after the specified starting file sequence number.

If a specific LABEL identifier is specified, it is compared with the label identifier of the data file specified by the starting sequence number. If the identifiers do not match, an error message is sent.

### Single values

**\*ALL** All data file labels on the tape are displayed.

### Element 1: Starting file sequence number

#### \*FIRST

The data file labels being displayed begin with the first file (or the only file) on the tape.

#### *1-16777215*

Specify the starting file sequence number. If the specified starting file sequence number is not found on the tape, an error message is sent.

## Element 2: Ending file sequence number

### \*LAST

The range of data file labels being displayed begins with the specified starting file sequence number and ends with the last data file label on tape.

### \*ONLY

Only the file specified in the starting sequence is displayed.

### 1-16777215

Specify the ending file sequence number. The specified ending file sequence number must be greater than or equal to the starting file sequence number.

**Note:** The ending file sequence number is ignored for OUTPUT(\*) for an interactive job. The user has control and can continue to display information or end whenever they desire.

**Note:** The ending file sequence number is ignored if a specific LABEL identifier is specified.

Top

---

## Data type (DATA)

Specifies the type of information that is shown.

### \*LABELS

The volume label and data file labels are shown.

### \*SAVRST

Shows the summary information about the command and each saved object.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*     The output is displayed (if requested by an interactive job) or printed with the job's spooled  
-     output (if requested by a batch job).

### \*PRINT

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

### \*OUTFILE

The output is directed to the database file specified on the OUTFILE parameter.

Top

---

## End of tape option (ENDOPT)

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

### \*REWIND

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

### \*UNLOAD

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

---

## File to receive output (OUTFILE)

Specifies the qualified name of the database file to which the information about the tape files is directed. 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 QATADOF in QSYS as a model with the format name of QTADOUTF.

**Note:** This parameter is valid only when OUTPUT(\*OUTFILE) and DATA(\*LABELS) is specified.

### Qualifier 1: File to receive output

*name* Specify the name of the database 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 job is searched. If no library is specified as the current library for the job, the QGPL library is used.

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

Top

---

## Output member options (OUTMBR)

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

### Element 1: Member to receive output

#### **\*FIRST**

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

*name* Specify the name of the file member that is to receive the output. If a member name is specified and the member does not exist, the system creates it.

### Element 2: Replace or add records

#### **\*REPLACE**

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

**\*ADD** The system adds the new records at the end of the existing member.

Top

---

## Examples

```
DSPTAP  DEV(QTAPE2)  LABEL(*ALL)
```

This command displays the volume label and file labels on the tape volume that is on the tape device named QTAPE2.

Top



---

## Error messages

### \*ESCAPE Messages

**CPD375A**

Media error on save media.

**CPF3704**

Request ended; data management error occurred.

**CPF3743**

File cannot be restored, displayed, or listed.

**CPF3792**

Information not displayed. Error occurred.

**CPF3793**

Machine or ASP storage limit reached.

**CPF6708**

Command ended due to error.

**CPF6718**

Cannot allocate device &1.

**CPF6721**

Device &1 not a tape device.

**CPF6723**

File not found on volume &2 on device &1.

**CPF6724**

File label &5 not found on volume &2.

**CPF6745**

Device &1 not a media library device.

**CPF6751**

Load failure occurred on device &4.

**CPF6760**

Device &1 not ready.

**CPF6772**

Volume on device &1 cannot be processed.

**CPF9814**

Device &1 not found.

**CPF9825**

Not authorized to device &1.

**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.



---

## Display Tape Category (DSPTAPCGY)

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

Parameters  
Examples  
Error messages

The Display Tape Category (DSPTAPCGY) command allows the user to display the categories defined through the Create Tape Category (CRTTAPCGY) command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *PRINT, *OUTFILE	Optional, Positional 1
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

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting workstation, printed to an output file, or printed with the job's spooled output.

\*  
- The output is displayed if requested by an interactive job or printed with the job's spooled output if requested by a batch job.

#### \*PRINT

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

#### \*OUTFILE

The output is directed to a database file specified on the OUTFILE parameter. The file must have the same format as database file QATACOF.

Top

---

### File to receive output (OUTFILE)

Specifies the qualified name of the database file to which the information about the tape volumes is directed. 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 QATACOF in QSYS as a model with the format name of QTACOUTF. This parameter is valid only when OUTPUT(\*OUTFILE) is specified.

### Qualifier 1: File to receive output

*name* Specify the name of the database 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 job is searched. If no library is specified as the current library for the job, the QGPL library is used.

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

Top

---

## Output member options (OUTMBR)

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

### Element 1: Member to receive output

#### **\*FIRST**

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

*name* Specify the name of the file member that is to receive the output. If a member name is specified and the member does not exist, the system creates it.

### Element 2: Replace or add records

#### **\*REPLACE**

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

**\*ADD** The system adds the new records at the end of the existing member.

Top

---

## Examples

```
DSPTAPCGY  OUTPUT(*)
```

This command displays the user defined categories on this system to the workstation display.

Top

---

## Error messages

### **\*ESCAPE Messages**

#### **CPF67E4**

Library device function not successful

Top

---

## Display Tape Cartridge (DSPTAPCTG)

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

Parameters  
Examples  
Error messages

The Display Tape Cartridge (DSPTAPCTG) command displays the attributes of tape cartridges.

Top

---

### Parameters

Keyword	Description	Choices	Notes
DEV	Library device	<i>Name</i>	Required, Positional 1
CTG	Cartridge ID	Single values: *ALL Other values (up to 40 repetitions): <i>Character value</i>	Optional
CGY	Category	Single values: *SHARE400, *INSERT, *EJECT Other values: <i>Element list</i>	Optional
	Element 1: Category name	<i>Character value</i> , *ALL, *NOSHARE, *IPL, *NL, *SYSGEN, *CNV	
	Element 2: Category system	<i>Character value</i> , *CURRENT, *ALL	
OUTPUT	Output	*, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , *FIRST	
	Element 2: Replace or add records	*REPLACE, *ADD	

Top

---

### Library device (DEV)

Specifies the library device to be used. The device description must have been created previously on the system using the Create Device Media Library (CRTDEVMLB) command.

This is a required parameter.

*name* Specify the name of the library device.

Top

---

### Cartridge ID (CTG)

Specifies 1 to 40 cartridge identifiers to be displayed.

**Note:** The cartridge identifier should represent the external identifier if the library device has a bar code scanner to read external identifiers.

### Single values

**\*ALL** All tape cartridges in the device or in the category specified are displayed.

### Other values (up to 40 repetitions)

#### *generic-identifier*

Specify the generic name of the cartridge identifier. 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 cartridge identifiers with names that begin with the generic prefix. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete cartridge identifier.

#### *identifier*

Specify the cartridge identifier.

Top

---

## Category (CGY)

Specifies the category of tape cartridges to be shown.

### Single values

#### **\*SHARE400**

The cartridge identifiers in the \*SHARE400 category are displayed. A cartridge in this category can be shared with other systems attached to the same device.

#### **\*INSERT**

The cartridge identifiers in the \*INSERT category are displayed. A cartridge in this category has been placed in the library device, but its identifier has not been added to the system.

#### **\*EJECT**

The cartridge identifiers in the \*EJECT category are displayed. A cartridge in this category has had its identifier removed from the system and is no longer usable.

### Element 1: Category name

**\*ALL** All categories are searched for the cartridge identifiers specified in the CTG parameter and all are displayed.

#### **\*NOSHARE**

The cartridge identifiers in the \*NOSHARE category are displayed. A cartridge with this identifier cannot be shared with other systems.

#### **\*IPL**

The cartridge identifiers in the \*IPL category are displayed. A cartridge with this identifier can be used for an alternate IPL.

#### **\*NL**

The cartridge identifiers in the \*NL category are displayed. A cartridge with this identifier is used as a non-labeled tape.

#### **\*SYSGEN**

The cartridge identifiers in the \*SYSGEN category are displayed. If the library device is in \*SYSGEN mode, cartridges cannot be moved from the \*SYSGEN category.

#### **\*CNV**

The cartridge identifiers in the \*CNV category are displayed. A cartridge in this category is for use with the convenience station.

*character-value*

Specify the name of a user-defined category. This category name must have been created previously with the Create Tape Category (CRTTAPCGY) command.

**Element 2: Category system**

This element identifies the system to which the category belongs. The system name is obtained from the pending system name field of a Display Network Attributes (DSPNETA) command. If there is no pending system name, the current system name attribute is used.

\*\*\*\*\* Attention \*\*\*\*\*  
If the system name is changed, all category information associated with all tape cartridges in library devices are not valid.  
\*\*\*\*\*

**\*CURRENT**

The system currently executing the command.

**\*ALL** All system names that own categories available to the system running this command are used.

*character-value*

Specify the name of the system that the category belongs to.

Top

**Output (OUTPUT)**

Specifies whether the output from the command is displayed at the requesting workstation, printed to an output file, or printed with the job's spooled output.

**\*** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).  
**-**

**\*PRINT**

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

**\*OUTFILE**

The output is directed to a database file specified on the OUTFILE parameter. The file must have the same format as database file QSYS/QATAVOF.

Top

**File to receive output (OUTFILE)**

Specifies the qualified name of the database file to which the information about the tape volumes is directed. 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 QATAVOF in QSYS as a model with the format name of QTAVOLOF.

**Note:** This parameter is valid only when OUTPUT(\*OUTFILE) is specified.

**Qualifier 1: File to receive output**

*name* Specify the name of the database 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 job is searched. If no library is specified as the current library for the job, the QGPL library is used.

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

Top

---

## **Output member options (OUTMBR)**

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

### **Element 1: Member to receive output**

#### **\*FIRST**

The first member in the file receives the output. If this value is specified and the member does not exist, the system creates a member with the name of the file specified on the OUTFILE parameter.

*name* Specify the name of the file member that is to receive the output. If a member name is specified and the member does not exist, the system creates it.

### **Element 2: Replace or add records**

#### **\*REPLACE**

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

**\*ADD** The system adds the new records at the end of the existing member.

Top

---

## **Examples**

### **Example 1: Displaying the Tape Cartridges in the \*SHARE400 Category**

```
DSPTAPCTG  DEV(LIB01)  CGY(*SHARE400)  OUTPUT(*)
```

This command displays the attributes of all tape cartridges in the \*SHARE400 category on the work station display.

### **Example 2: Displaying the Tape Cartridge for VOL3**

```
DSPTAPCTG  DEV(LIB01)  CTG(VOL3)  OUTPUT(*)  CGY(*ALL)
```

This command displays the attributes of the cartridge identifier VOL3.

Top

---

## **Error messages**

### **\*ESCAPE Messages**

#### **CPF6708**

Command ended due to error.

#### **CPF6711**

Command not allowed



**CPF6718**  
Cannot allocate device &1.

**CPF6745**  
Device &1 not a media library device.

**CPF67A6**  
Category does not exist

**CPF67D2**  
Cartridge command was not successful.

**CPF67D4**  
Category not available

**CPF67E4**  
Library device function not successful

**CPF67E6**  
Volume &2 is not correct

**CPF67EA**  
Function not successful

**CPF67EC**  
Library device description &1 does not exist

**CPF67ED**  
Library device &1 not available

**CPF67F5**  
Duplicate cartridge or virtual volume name found

**CPF9814**  
Device &1 not found.

**CPF9825**  
Not authorized to device &1.

Top



---

## Display Tape Status (DSPTAPSTS)

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

Parameters  
Examples  
Error messages

The Display Tape Status (DSPTAPSTS) command does the following:

- Displays slot information associated with the media library device.
- Displays information about the resources attached to the media library device.

Top

---

### Parameters

Keyword	Description	Choices	Notes
DEV	Library device	Name, <u>*ALL</u>	Optional, Positional 1
OUTPUT	Output	*, <u>_</u> , *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	Qualified object name	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
OUTMBR	Output member options	Element list	Optional
	Element 1: Member to receive output	Name, <u>*FIRST</u>	
	Element 2: Replace or add records	<u>*REPLACE</u> , *ADD	
OUTFILFMT	Outfile format	<u>*TYPE1</u> , *TYPE2	Optional

Top

---

### Library device (DEV)

Specifies the name of the media library device in which information is displayed.

\*ALL Specifies that all tape media library devices defined through Create Device Media Library (CRTDEVMLB) command are displayed.

*name* Specify the device name. The device name must be a tape media library device name or a random access cartridge loader (RACL) device name. The device name must already be known to the system by Create Device Media Library (CRTDEVMLB) command.

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting workstation, printed to an output file, or printed with the job's spooled output.

\*  
- The requested output is shown on the display. If this value is specified for a batch job, the effect is the same as if \*PRINT were entered.

### **\*PRINT**

The requested output is written to a spooled file, which is found in the job's output queue.

### **\*OUTFILE**

The output is directed to a database file specified on the OUTFILE parameter. The file must have the same format as database file QATAIOF or QATAIOF2.

Top

---

## **File to receive output (OUTFILE)**

Specifies the database file to which the information about the tape media libraries is directed. If the file does not exist, this command creates a database file in the specified library. If a new file is created with `OUTFILFMT(*TYPE1)`, the system uses QATAIOF in QSYS as a model with the format name of QTAIOUTF. If a new file is created with `OUTFILFMT(*TYPE2)`, the system uses QATAIOF2 in QSYS as a model with the format name of QTAIOF2. This parameter is valid only when `OUTPUT(*OUTFILE)` is specified.

### **Qualifier 1: File to receive output**

*name* Specify the name of the database 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 job is searched. If no library is specified as the current library for the job, the QGPL library is used.

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

Top

---

## **Output member options (OUTMBR)**

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

### **Element 1: Member to receive output**

#### **\*FIRST**

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

*name* Specify the name of the file member that is to receive the output. If a member name is specified and the member does not exist, the system creates it.

### **Element 2: Replace or add records**

#### **\*REPLACE**

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

**\*ADD** The system adds the new records at the end of the existing member.

Top

---

## Output member options (OUTFILFMT)

Specifies the format of the output file when OUTPUT(\*OUTFILE) is specified.

### \*TYPE1

The format is defined by model output file QATAIOF in library QSYS with record format name QTAIOUTF.

### \*TYPE2

The format is defined by model output file QATAIOF2 in library QSYS with record format name QTAIOF2.

Top

---

## Examples

```
DSPTAPSTS  DEV(LIB01)  OUTPUT(*)
```

This command displays the valid information about this library device to the workstation display.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF6718

Cannot allocate device &1.

#### CPF6745

Device &1 not a media library device.

#### CPF67E4

Library device function not successful

#### CPF67E7

No library devices exist

#### CPF9814

Device &1 not found.

#### CPF9825

Not authorized to device &1.

#### CPF9845

Error occurred while opening file &1.

#### CPF9846

Error while processing file &1 in library &2.

#### CPF9847

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

#### CPF9860

Error occurred during output file processing.

#### CPF9871

Error occurred while processing.

Top



---

## Display Trademarks (DSPTM)

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

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

The Display Trademarks (DSPTM) command displays a list of trademarks that appear in the names of licensed products.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

DSPTM

This command displays a list of trademarks.

[Top](#)

---

### Error messages

None

[Top](#)





---

## Display Trace (DSPTRC)

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

Parameters  
Examples  
Error messages

The Display Trace (DSPTRC) command shows all of the traces that are currently defined in the programs specified in this command. The following trace information is shown:

- The statement ranges or machine instruction ranges in the program
- The name or MI ODV numbers of all the program variables associated with the trace statements
- If the variables are recorded whenever the trace statement is processed or only when their values are changed

**Restriction:** This command is valid only in debug mode.

Top

---

## Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *PRINT _	Optional, Positional 1
PGM	Program	Single values: *DFTPGM, *ALL Other values (up to 20 repetitions): <i>Name</i>	Optional

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_ The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

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

Top

---

## Program (PGM)

Specifies which programs in debug mode have their trace statements and associated program variables shown.

**\*DFTPGM**  
Only the default program has its trace statements shown.

**\*ALL** All the programs currently in debug mode have their trace statements shown.

***name*** Specify the names of up to 20 programs to have their trace statements shown. The programs specified must already be in debug mode.

---

## Examples

DSPTRC

This command shows all of the trace data statement ranges currently specified in the default program of this debugging session. Also displayed are the program variables (but not their values) that are associated with the trace data statements.

Top

---

## Error messages

### \*ESCAPE Messages

CPF1999

Errors occurred on command.

Top

---

## Display Trace Data (DSPTRCDTA)

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

Parameters  
Examples  
Error messages

The Display Trace Data (DSPTRCDTA) command shows the output of any traces performed since the most recent Clear Trace Data (CLRTRCDTA) command. All of the trace statements and associated program variables within the trace range are shown. The display shows the sequence in which the traced statements or machine instructions were processed and the name or machine-interface object-definition-table-vector (MI ODV) number and value of any program variables defined for the trace at each point in the sequence. Note that the display of variable values is controlled by the **When output (OUTVAR)** parameter on the Add Trace (ADDTRC) command that defined the trace being shown.

If a job is in debug mode, and that job ends before an End Debug (ENDDDBG) is done, this command is done automatically, printing the output with the job's spooled output.

**Note:** A program statement appears in the trace data prior to its processing a variable. Therefore, when a program statement (Statement A) changes a variable specified in the trace, the new value appears in the trace data after the statement that follows Statement A.

**Restriction:** This command is valid only in debug mode.

Top

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *_	Optional, Positional 1
CLEAR	Clear	*NO, *YES	Optional

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_ The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

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

Top

---

### Clear (CLEAR)

Specifies whether the trace data is cleared after it has been shown.

\*NO The trace data is not cleared.

\*YES The trace data is cleared after it has been shown.

Top

---

## Examples

DSPTRCDTA

This command shows all of the recorded trace data at the requesting display station. All of the trace statements in the trace range and the values of the associated program variables are displayed. The trace data is not cleared after it has been displayed because CLEAR(\*NO) is assumed.

Top

---

## Error messages

### \*ESCAPE Messages

CPF1999

Errors occurred on command.

Top

---

## Display User-Defined FS (DSPUDFS)

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

Parameters  
Examples  
Error messages

The Display User-Defined File System (DSPUDFS) command displays the attributes and, optionally, the extended attributes for an existing user-defined file system (UDFS).

Top

---

### Parameters

Keyword	Description	Choices	Notes
UDFS	User-defined file system	<i>Path name</i>	Required, Positional 1
OUTPUT	Output	*, *PRINT _	Optional

Top

---

### User-defined file system (UDFS)

Specifies the path name of the file system to be displayed. It must be (or resolve to a pathname) of one of the following two forms:

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

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

This is a required parameter.

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_ The output is displayed for interactive jobs or printed with the job's spooled output for non-interactive jobs.

**\*PRINT**

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

Top

---

## Examples

### Example 1: Displaying a User-defined File System

```
DSPUDFS  UDFS('/dev/QASP05/joe.udfs')
```

This command displays the attributes of a user-defined file system (UDFS) named *joe* in the user auxiliary storage pool (ASP) 5.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPFA0A9

Object not found. Object is &1.

#### CPFA09C

Not authorized to object. Object is &1.

[Top](#)

---

## Display Upgrade Preparation (DSPUPGPRP)

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

Parameters  
Examples  
Error messages

The Display Upgrade Preparation (DSPUPGPRP) command displays upgrade preparation information when upgrading to a new release of OS/400.

### Restrictions:

1. The STRUPGPRP command must have completed successfully and file QUSRSYS/QAIZADSK must exist or an error message is issued.

Top

---

## Parameters

Keyword	Description	Choices	Notes
TYPE	Type of information	*ALL, *OBJ, *PRB, *STG, *DSK, *SFW, *HDW, *OWN, *ORD, *TIMEEST	Optional, Positional 1
OUTPUT	Output	*, *PRINT _	Optional, Positional 2

Top

---

## Type of information (TYPE)

Specifies the type of preparation information to include in the report.

**\*ALL** All preparation information is displayed.

**\*OBJ** Information about unsupported objects is displayed.

**\*PRB** Information about potential problems is displayed.

**\*STG** Information about disk storage estimates is displayed.

**\*SFW** Information about installed licensed programs is displayed.

**\*HDW**  
Information about hardware resources is displayed.

**\*ORD** Recommendations for your upgrade order are displayed.

**\*OWN**  
Information about objects and their owners is displayed for use in cleaning up system.

**\*DSK** Disk configuration information is displayed.

**\*TIMEEST**  
Downtime estimates are displayed.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting workstation or printed with the job's spooled output.

\*  
\_ The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

### \*PRINT

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

Top

---

## Examples

None

Top

---

## Error messages

Unknown

Top



---

## Display User Permission (DSPUSRPMN)

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

Parameters  
Examples  
Error messages

The Display User Permission (DSPUSRPMN) command allows you to show which users are permitted to handle documents or folders and perform OfficeVision/400-related tasks on behalf of another user, or to show which users have permitted other users to work on their behalf.

Top

---

### Parameters

Keyword	Description	Choices	Notes
USER	User profile	Single values: *CURRENT, *ALL Other values (up to 300 repetitions): <i>Name</i>	Optional, Positional 1
GRANTED	Granted to/from	<i>Character value</i> , *TO, *FROM	Optional
OUTPUT	Output	*, *PRINT	Optional

Top

---

### User profile (USER)

Specifies the name of the user profile for which the information is shown. The name on this parameter must be enrolled in the system distribution directory before you run this command.

#### \*CURRENT

Your user permission information is shown.

**\*ALL** Information is shown for all users in the information directory.

*name* Specify the name of the user profile for which information is shown.

Top

---

### Granted to/from (GRANTED)

Specifies whether the output produced should display the granted-to relationships or granted-from relationships of a user or users.

**\*TO** Users who are permitted to work on behalf of the specified user are shown.

#### **\*FROM**

Users that have permitted the specified user to work on their behalf are shown.

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_ The output is displayed at the requesting work station if requested by an interactive job. If this is not an interactive job, the output is printed with the job's spooled output.

**\*PRINT**

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

Top

---

## Examples

### Example 1: Displaying Names of Users

```
DSPUSRPMN  USER(*CURRENT)  GRANTED(*TO)  OUTPUT(*PRINT)
```

This command prints the names of the users who are permitted to work on behalf of the current user.

### Example 2: Printing Names of Users

```
DSPUSRPMN  USER(*ALL)  GRANTED(*TO)  OUTPUT(*PRINT)
```

This command prints the names of all users who have permitted other users to work on their behalf as well as the names of the other users.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9043

User permission is not displayed because error occurred.

#### 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.

Top

---

## Display User Profile (DSPUSRPRF)

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

Parameters  
Examples  
Error messages

The Display User Profile (DSPUSRPRF) command displays the contents of a user profile. The user profile contains the user's operational limits for system resources. This includes the names of the objects, commands, and devices that the user has specific authority to use and the names of the objects that the user owns, and that the user is the primary group for.

Objects owned by the user profile are not shown on the \*CMDAUT, \*DEVAUT, \*OBJAUT, or \*OBJPGP displays.

This command does not show the password, nor does it show information about objects authorized for public use. The document password is not shown on the \*BASIC display or on any CL command output. Any user on the system can be authorized to use the DSPUSRPRF command, but the requesting user must have read (\*READ) authority for the user profile being displayed.

The DSPUSRPRF function may be a long-running function, depending upon the number of objects the user profile owns and is authorized to use.

**Restrictions:** The user name can be specified as USRPRF(\*ALL) or USRPRF(generic-name) only when TYPE(\*BASIC) and OUTPUT(\*OUTFILE) are specified.

Top

---

### Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	Generic name, name, *ALL	Required, Positional 1
TYPE	Type of information	*BASIC, *ALL, *CMDAUT, *DEVAUT, *GRPMBR, *OBJAUT, *OBJOWN, *OBJPGP	Optional, Positional 2
OUTPUT	Output	*, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	Qualified object name	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
OUTMBR	Output member options	Element list	Optional
	Element 1: Member to receive output	Name, *FIRST	
	Element 2: Replace or add records	*REPLACE, *ADD	

Top

---

## User profile (USRPRF)

Specifies the user profiles to be displayed.

This is a required parameter.

**\*ALL** All user profiles are shown.

### *generic-name*

Specify the generic name of the user profiles to be shown. A generic name is a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, all user profiles that have names with the same prefix as the generic name are shown.

**name** Specify the name of the user profile to be shown.

**Note:** \*ALL or a generic name can only be specified when TYPE(\*BASIC) and OUTPUT(\*OUTFILE) are specified.

Top

---

## Type of information (TYPE)

Specifies the types of information that can be displayed. All, or one, of the following can be displayed:

- The basic portion of the user profile that describes the user
- Commands for which the user profile has specific authority
- Devices for which the user profile has specific authority
- All objects (including commands and devices) for which the user has some specific authority and the authorities assigned with those objects
- Objects that are owned by the user
- Objects that the user is the primary group for.
- Members of the group, if the user profile is a group profile

### **\*BASIC**

All parameters as defined in the user profile, are displayed.

**\*ALL** All of the information in the user profile is displayed.

### **\*CMDAUT**

Displays the control language commands to which the user has specific authority.

### **\*DEVAUT**

Displays the system devices to which the user has specific authority.

### **\*OBJAUT**

Displays the names of the objects (except those authorized for public use) to which the user has specific authority, the user's authority for those objects, and the object types.

### **\*OBJOWN**

Displays the total number of objects this user owns, the object names, the object types, and the libraries in which the objects reside. Also indicates if the object is an authority holder.

### **\*OBJPGP**

Displays the total number of objects the user is the primary group for, the object names, the type, the library the object resides in, and the primary group authority. Also indicates if the object is an authority holder.

### **\*GRPMBR**

Displays the members of a group. This display is available only if the user profile being displayed is a group profile.

---

## Output (OUTPUT)

Specifies where the output from the command is sent.

\*  
- The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

### \*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.

---

## 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 and \*BASIC is specified on the **Type of information (TYPE)** TYPE parameter, the system uses QADSPUPB in QSYS with a format name QSYDSUPB as a model.

If a new file is created and \*OBJAUT is specified on the TYPE parameter, the system uses QADSPUPA in QSYS with a format name QSYDSUPA as a model.

If a new file is created and \*OBJOWN is specified on the TYPE parameter, the system uses QADSPUPO in QSYS with a format name QSYDSUPO as a model.

If a new file is created and \*OBJPGP is specified on the TYPE parameter, the system uses QADSPUPG in QSYS with a format name QSYDSUPG as a model.

---

## Output member options (OUTMBR)

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

### Element 1: Member to receive output

#### \*FIRST

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

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

### Element 2: Replace or add records

#### \*REPLACE

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

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

Top

---

## Examples

### Example 1: Displaying Basic Information

```
DSPUSRPRF  USRPRF(THSMITH)
```

This command shows the basic portion of the user profile named THSMITH because TYPE(\*BASIC) is assumed. The commands, devices, and objects that the user is authorized to use are not displayed. Because OUTPUT(\*) is also assumed, the operational information is either displayed or printed, depending on where the command is submitted.

### Example 2: Printing a List of Objects

```
DSPUSRPRF  USRPRF(RTJOHNSON)  TYPE(*OBJOWN)  OUTPUT(*PRINT)
```

This command causes the list of objects that are owned by the user named RTJOHNSON to be printed. The list contains the object names, object types, and the names of the libraries where the objects are located.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF22DF

Unable to process request for user profile &1.

#### CPF22D8

Use of generic user profile name not correct.

#### CPF22D9

No user profiles of specified name exist.

#### CPF22EB

Unable to process request for user profile &1.

**CPF2204**

User profile &1 not found.

**CPF2213**

Not able to allocate user profile &1.

**CPF2217**

Not authorized to user profile &1.

**CPF2257**

User profile &1 not a group profile.

**CPF9860**

Error occurred during output file processing.

[Top](#)





---

## Display User Print Info (DSPUSRPTI)

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

Parameters  
Examples  
Error messages

The Display User Print Information (DSPUSRPTI) command displays the user print information for the specified user profile.

Top

---

### Parameters

Keyword	Description	Choices	Notes
USER	User	<i>Name</i> , *CURRENT	Optional, Positional 1
OUTPUT	Output	*, *_PRINT	Optional, Positional 2

Top

---

### User (USER)

Specifies the name of the user whose print information is to be displayed.

#### \*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 to be displayed.

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting work station or printed with the job's spooled output. More information on this parameter is in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter> book.

\*  
- Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

#### \*PRINT

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

Top

---

### Examples

```
DSPUSRPTI USER(FEIST)
```

This command displays the user print information for user profile FEIST.

---

## 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.

---

## Display Work Station User (DSPWSUSR)

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

Parameters  
Examples  
Error messages

The Display Work Station User (DSPWSUSR) command allows you to display information about your current session. Information shown includes:

- Display station
- Number of secondary interactive jobs in session
- Secondary interactive job currently active

Top

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *_PRINT _	Optional, Positional 1

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_     The output is displayed (if requested by a secondary interactive job) or printed with the job's spooled output (if requested by a batch job).

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

Top

---

### Examples

DSPWSUSR   OUTPUT(\*)

This command displays the information describing the current job.

Top

---

### Error messages

None

Top



---

## Duplicate Optical (DUPOPT)

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

Parameters  
Examples  
Error messages

The Duplicate Optical (DUPOPT) command creates a duplicate optical volume. The duplicate volume is identical to the original volume except for the volume identifier and the time it was created.

**Restriction:** To use this command you must have \*USE authority to the authorization list securing the source volume. You need \*ALL authority to the authorization list securing the target volume if it is in an optical media library. You need \*CHANGE authority to the authorization list securing the target volume if it is in an optical device.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FROMVOL	From volume identifier	Character value, *MOUNTED	Required, Positional 1
TOVOL	To volume identifier	Character value, *MOUNTED	Required, Positional 2
NEWVOL	New volume identifier	Character value, *TOVOL, *FROMVOL	Optional, Positional 3
CLEAR	Clear	*NO, *YES	Optional
FROMDEV	From device	Name	Optional
TODEV	To device	Name	Optional
ALWMEDERR	Allow media errors	*NONE, *FILE	Optional
OUTPUT	Output	*ERROR, *NONE	Optional
FROMENDOPT	From end of media option	*LEAVE, *UNLOAD	Optional
TOENDOPT	To end of media option	*LEAVE, *UNLOAD	Optional

Top

---

### From volume identifier (FROMVOL)

Specifies the volume identifier of the optical volume being duplicated. To determine the volume identifier of media not in an optical media library issue the following Display Optical CL command: DSPOPT VOL(\*MOUNTED) DEV(device).

#### *from-volume-identifier*

Specify the source volume identifier.

#### \*MOUNTED

Use the optical volume mounted in the optical device specified by the FROMDEV parameter.

**Note:** This value is not valid for volumes in optical media library devices.

Top

---

## To volume identifier (TOVOL)

Specifies the volume identifier of the optical volume being created (new volume). This volume must have the same physical characteristics as the volume specified on the FROMVOL parameter, but cannot be the volume on the opposite side of the cartridge. If the volume specified is WORM (write-once-read-many) media, it must be a volume that is not initialized. Use the Display Optical (DSPOPT) CL command to determine the physical characteristics of an optical volume.

### *to-volume-identifier*

Specify the target volume identifier.

### **\*MOUNTED**

Use the optical volume mounted in the optical device specified by the TODEV parameter.

**Note:** This value is not valid for volumes in optical media library devices.

Top

---

## New volume identifier (NEWVOL)

Specifies the new volume identifier of the to-volume after the duplication is complete.

### **\*TOVOL**

The volume identifier will be identical to the to-volume identifier.

### **\*FROMVOL**

The volume identifier will be identical to the from-volume identifier. This parameter is only allowed for volumes that are not in a library device and either FROMENDOPT or TOENDOPT is \*UNLOAD.

### *new-volume-identifier*

Specify the new volume identifier of the to-volume.

Top

---

## Clear (CLEAR)

Indicates whether to re-initialize the target optical volume if the volume is found to be already initialized.

**Note:** This parameter is ignored if the volume is WORM (write-once-read-many) media.

**\*NO** The volume is not re-initialized.

**\*YES** The volume is re-initialized.

**Note:** For media type \*ERASE, specifying \*YES will result in all existing data being erased prior to the start of the duplication process.

For media type \*DVD-RAM, specifying \*YES will not result in existing data being erased prior to the start of the duplication process. Though the data is not erased, access to the data is lost. If it is required that data on \*DVD-RAM media be erased, initialize the volume using Initialize Optical (INZOPT) prior to running DUPOPT. Specify the CLEAR(\*YES) parameter on the Initialize Optical (INZOPT) CL command.

Top

---

## From device (FROMDEV)

Specifies the optical device which contains the from-volume.

**Note:** This parameter is only required if parameter FROMVOL is specified as \*MOUNTED.

### *from-optical-device*

Specify the name of the optical device containing the from-volume.

Top

---

## To device (TODEV)

Specifies the optical device which contains the to-volume.

**Note:** This parameter is only required if parameter TOVOL is specified as \*MOUNTED.

### *to-optical-device*

Specify the name of the optical device containing the to-volume.

Top

---

## Allow media errors (ALWMEDERR)

Specifies whether or not the Duplicate Optical (DUPOPT) command should terminate if an error is encountered while trying to duplicate the file data areas.

### \*NONE

When an error is encountered duplicating the file data, stop the duplication and return the error.

**\*FILE** When an error is encountered duplicating the file data, continue the duplication. A list of files not duplicated may be output by specifying OUTPUT(\*ERROR).

Top

---

## Output (OUTPUT)

Specifies whether or not the output from the command is printed with the job's spooled output.

**Note:** This parameter is valid only when \*FILE is specified for the **Allow media errors (ALWMEDERR)** parameter.

### \*ERROR

If all of the files are duplicated, there is no output. If there are files that were not duplicated, a list of these files will be printed with the job's spooled output. Escape message OPT2047 will also be returned.

### \*NONE

If all of the files are duplicated, there is no output. If there are files that were not duplicated, the number of files not duplicated will be in the escape message OPT2047.

Top

---

## From end of media option (FROMENDOPT)

After the DUPOPT request has completed, specifies whether to leave or unload the from-volume from the optical device in which it is located.

**Note:** This parameter is ignored if the from-volume is in an optical media library device.

### \*LEAVE

The from-volume is left in the optical device.

### \*UNLOAD

The from-volume is unloaded from the optical device.

Top

---

## To end of media option (TOENDOPT)

After the DUPOPT request has completed, specifies whether to leave or unload the to-volume from the optical device in which it is located.

**Note:** This parameter is ignored if the to-volume is in an optical media library device.

### \*LEAVE

The to-volume is left in the optical device.

### \*UNLOAD

The to-volume is unloaded from the optical device.

Top

---

## Examples

### Example 1: Duplicate an Optical Volume When the Volume Names are Known.

```
DUPOPT FROMVOL(VOL01) TOVOL(VOL02) NEWVOL(*TOVOL)
      CLEAR(*YES)
```

This command creates a duplicate of the optical volume VOL01 on volume VOL02, which keeps the same volume identifier. VOL02 will be re-initialized prior to the duplication process.

### Example 2: Duplicate an Optical Volume When the Device Names are Known.

```
DUPOPT FROMVOL(*MOUNTED) TOVOL(*MOUNTED) NEWVOL(BKP001)
      CLEAR(*YES) FROMDEV(OPT01) TODEV(OPT02)
      FROMENDOPT(*LEAVE) TOENDOPT(*UNLOAD)
```

This command duplicates the optical volume in optical device OPT01 onto the volume in device OPT02. The optical volume in device OPT02 is re-initialized prior to the duplication process. The volume in device OPT01 will be left in the device after the duplication process completes. The volume in device OPT02 will be unloaded after the duplication process completes and will have a volume identifier of BKP001.

### Example 3: Duplicate an Optical Volume and Continue the Duplication if Damaged Files are Found.

```
DUPOPT FROMVOL(VOL01) TOVOL(VOL02) NEWVOL(*TOVOL)
      CLEAR(*YES) ALWMEDEERR(*FILE) OUTPUT(*ERROR)
```

This command creates a duplicate of the optical volume VOL01 on volume VOL02, which keeps the same volume identifier. VOL02 will be re-initialized prior to the duplication process. If there are any damaged



files on VOL01 the duplication will continue and the damaged files will not be duplicated. The names of the damaged files will be printed to the job's spooled output.

**Example 4: Duplicate an Exact Copy of a Volume Using NEWVOL(\*FROMVOL).**

```
DUPOPT FROMVOL(VOL01) TOVOL(VOL02) NEWVOL(*FROMVOL)
      CLEAR(*YES) TODEV(OPT02) TOENDOPT(*UNLOAD)
```

This command creates a duplicate of the optical volume VOL01 on volume VOL02. VOL02 will be re-initialized prior to the duplication process. The volume in device OPT02 will be unloaded after the duplication process completes and will have a volume identifier of VOL01.

Top

---

## Error messages

### \*ESCAPE Messages

**OPT1305**

Optical volume &1 is read only.

**OPT1315**

Optical volume &1 is write protected.

**OPT1320**

Optical volume &1 in use.

**OPT1325**

Optical volume format not recognized.

**OPT1330**

Optical volume not found or not useable.

**OPT1331**

Optical volume &1 not found.

**OPT1335**

Volume &1 already initialized.

**OPT1338**

Operation not supported for library device.

**OPT1340**

Optical volume &1 not initialized.

**OPT1342**

Invalid volume identifier specified.

**OPT1346**

Operation not allowed to volume located in a remote optical device.

**OPT1350**

Write operation failed to optical volume &1.

**OPT1375**

Optical volume &1 already exists.

**OPT1460**

Optical volume &1 is not in an optical device.

**OPT1499**

Source and target volumes are in different optical device types.

- OPT1515**  
Unsupported or insufficient configuration on optical device &1.
- OPT1530**  
&1 does not represent a valid optical device.
- OPT1555**  
Optical device &1 in use.
- OPT1605**  
Media or device error occurred.
- OPT1790**  
Operation not allowed or conflicts with another request.
- OPT1805**  
Error accessing optical volume index file.
- OPT1810**  
Error accessing optical directory index file.
- OPT1815**  
Internal program error occurred.
- OPT1820**  
Internal error occurred on optical device &1.
- OPT1821**  
Error occurred on optical device &1.
- OPT1825**  
Optical indexes are incorrect for optical device &1.
- OPT1860**  
Request to optical device &1 failed.
- OPT1861**  
No device description configured for resource &1.
- OPT1862**  
No active device description for resource &1.
- OPT1863**  
Optical libraries need to be reclaimed.
- OPT1864**  
Insufficient allocated and operational optical drives.
- OPT1872**  
Optical request timed out or was cancelled.
- OPT2029**  
TOVOL on opposite side of FROMVOL
- OPT2047**  
Duplicate Optical completed. &3 files were not duplicated.
- OPT2050**  
The duplicate optical volume request from optical volume &1 to optical volume &2 failed.
- OPT2301**  
Internal system object in use.
- OPT2420**  
Not authorized to optical volume &2.

**OPT7740**

User not authorized to object &2 in library &3 type &4.

Top



# Duplicate Tape (DUPTAP)

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

Parameters  
 Examples  
 Error messages

The Duplicate Tape (DUPTAP) command copies the contents of one tape to another tape.

## Notes:

1. The density field in the file header labels are updated to reflect the true density.
2. Byte 80 in the volume label of a tape written on device type 6157 is reset from a 'Q' to a blank.

## Restrictions:

- You must have two tape drives or a tape media library device with two tape resources to use this command.
- A file that spans volumes must have both partial files duplicated at the same time. That is, duplicating a tape that ends in a partial file, followed by appending the second part of the file to the end of the tape is not allowed. You must duplicate both parts of the file at the same time by specifying multiple volumes on the FROMVOL parameter.

Top

## Parameters

Keyword	Description	Choices	Notes
FROMDEV	From device	<i>Name</i>	Required, Positional 1
TODEV	To device	<i>Name</i>	Required, Positional 2
FROMSEQNBR	From sequence number	Single values: *ALL Other values: <i>Element list</i>	Optional
	Element 1: Starting file sequence number	1-16777215, *FIRST	
	Element 2: Ending file sequence number	1-16777215, *ONLY, *LAST	
TOSEQNBR	To sequence number	1-16777215, *END, *FROMSEQ	Optional
FROMVOL	From volume identifier	Single values: *MOUNTED Other values (up to 300 repetitions): <i>Character value</i>	Optional
TOVOL	To volume identifier	Single values: *MOUNTED, *FROMVOL Other values (up to 300 repetitions): <i>Character value</i>	Optional
TODENSITY	Tape density	<i>Character value</i> , *DEVTYPE, *CTGTYPE, *FMT3480, *FMT3490E, *FMT3570, *FMT3570E, *FMT3590, *FMT3590E, *FMT3590H, *QIC120, *QIC525, *QIC1000, *QIC2GB, *QIC2DC, *QIC4GB, *QIC4DC, *QIC3040, *QIC5010, *MLR3, *SLR60, *SLR100, *FMT2GB, *FMT5GB, *FMT7GB, *FMT20GB, *FMT60GB, *ULTRIUM1, *ULTRIUM2, *VXA1, *VXA2, 1600, 3200, 6250	Optional
COMPACT	Data compaction	*FROMFILE, *YES, *NO	Optional
FILES	Files to duplicate	*ALL, *ACTIVE	Optional

Keyword	Description	Choices	Notes
USRLBLPGM	User label program	Single values: *NONE, *SYSCOPY Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: User label program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
FROMENDOPT	From device end option	*REWIND, *UNLOAD, *LEAVE	Optional
TOENDOPT	To device end option	*UNLOAD, *REWIND, *LEAVE	Optional
CHECK	Check for active files	*YES, *NO	Optional
EXPDATE	File expiration date	<i>Date</i> , *FROMFILE, *PERM	Optional

Top

---

## From device (FROMDEV)

Specifies the device from which the tape is copied.

This is a required parameter.

**name** Specify the name of the tape device where the tape to be copied is mounted.

Top

---

## To device (TODEV)

Specifies the device to which the tape is copied.

This is a required parameter.

**name** Specify the name of the tape device where the tape volume to which data is being copied is mounted.

Top

---

## From sequence number (FROMSEQNBR)

Specifies which data file sequence numbers are to be copied.

### Single values

\*ALL All files are duplicated.

### Element 1: Starting file sequence number

#### \*FIRST

All files starting with the first file sequence are duplicated.

#### 1-16777215

Specify the starting file sequence number to be duplicated. Only the files in the specified sequence number range are duplicated.

### Element 2: Ending file sequence number

#### \*LAST

All files ending with the last file sequence are duplicated.

**\*ONLY**

Only the file specified in the starting file sequence is duplicated.

**1-16777215**

Specify the ending file sequence number of the range to be duplicated.

Top

---

## To sequence number (TOSEQNBR)

Specifies which sequence number the data files are to be copied to.

**\*FROMSEQ**

The data files are duplicated to the same file sequences as are specified in the from-file sequence number parameter.

**\*END** The data files are added to the logical end of tape. The next valid sequence number is used.

**1-16777215**

Specify the sequence number in which the data file will be copied to. This value is not allowed if the device does not have overwriting capabilities and the value specified is not the next logical value to be used at the end of the logical tape volume. The duplication begins at the specified file.

Top

---

## From volume identifier (FROMVOL)

Specifies the volume identifier of the tape being duplicated.

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

### Single values

**\*MOUNTED**

Any labeled or unlabeled volume placed on the tape device specified on the **From device (FROMDEV)** parameter is duplicated. For a tape media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command. For a virtual tape device, the volume to be used is the currently mounted one, or if there is not a currently mounted volume, the next volume in loaded status in the image catalog will be used.

### Other values (up to 300 repetitions)

*character-value*

Specify the identifier of the labeled volume being duplicated.

Top

---

## To volume identifier (TOVOL)

Specifies the volume identifiers of the tapes to which data is being copied.

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

### Single values

### \*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 category mounted by the Set Tape Category (SETTAPCGY) command. For a virtual tape device, the volume to be used is the currently mounted one, or if there is not a currently mounted volume, the next volume in loaded status in the image catalog will be used.

### \*FROMVOL

The volume label of the tape placed in the device specified on the **From device (FROMDEV)** parameter is used to initialize the tape placed in the device specified on the **To device (TODEV)** parameter. Up to eight additional volume labels and nine user volume labels are duplicated. This value is not supported for tape media library devices and virtual tape devices.

### **Other values (up to 300 repetitions)**

#### *character-value*

Specify the volume identifier of the tapes to which data is being copied. At the end of volume time, you are able to reinitialize the tape using this volume identifier. If the volume contains the correct volume identifier but is in the wrong code or density, the tape is reinitialized to the correct code and density. The volume identifier is saved.

Top

---

## **Tape density (TODENSITY)**

Specifies the density or format in which the copied data is written.

### \*DEVTYPE

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

#### **Device**

#### **Highest capacity density or format**

3480 \*FMT3480

3490E \*FMT3490E

#### **3570-Bxx**

\*FMT3570

#### **3570-Cxx**

\*FMT3570E

#### **3580-001**

\*ULTRIUM1

#### **3580-002**

\*ULTRIUM2

#### **3580-003**

\*ULTRIUM3

#### **3590-Bxx**

\*FMT3590

#### **3590-Exx**

\*FMT3590E

#### **3590-Hxx**

\*FMT3590H



3592-J1A  
\*FMT3592A1

4685-001  
\*VXA2

6335 \*QIC3040

6343 \*QIC1000

6344 \*QIC2GB

6348 \*QIC1000

6349 \*QIC2GB

6368 \*QIC1000

6369 \*QIC2GB

6379 \*QIC1000

6380 \*QIC2GB

6381 \*QIC2DC

6382 \*QIC4DC

6383 \*QIC5010

6384 \*SLR60

6385 \*QIC5010

6386 \*MLR3

6387 \*SLR100

6390 \*FMT7GB

63B0 \*VRT256K

7207-122  
\*QIC4DC

7208-002  
\*FMT2GB

7208-012  
\*FMT5GB

7208-222  
\*FMT7GB

7208-342  
\*FMT20GB

7208-345  
\*FMT60GB

9348 6250

**\*CTGTYPE**

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

*character-value*

Specify the density or format to use.

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

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

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

**\*FMT3480**

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

**\*FMT3490E**

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

**\*FMT3570**

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

**\*FMT3570E**

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

**\*FMT3590**

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

**\*FMT3590E**

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

**\*FMT3590H**

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

**\*FMT3592A1**

The format of this tape is FMT3592A1. The data format is written on the tape volume with a 3592 device. This density is used for 1/2 inch cartridge tapes.

**\*QIC120**

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

**\*QIC525**

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

**\*QIC1000**

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

**\*QIC2GB**

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

**\*QIC2DC**

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

**\*QIC4GB**

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

**\*QIC4DC**

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

**\*QIC3040**

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

**\*QIC5010**

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

**\*MLR3**

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

**\*SLR60**

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

**\*SLR100**

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

**\*FMT2GB**

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

**\*FMT5GB**

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

**\*FMT7GB**

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

**\*FMT20GB**

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

**\*FMT60GB**

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

**\*ULTRIUM1**

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

**\*ULTRIUM2**

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

**\*ULTRIUM3**

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

**\*VRT32K**

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

**\*VRT64K**

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

**\*VRT240K**

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

**\*VRT256K**

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

**\*VXA1**

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

**\*VXA2**

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

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

Top

---

## Data compaction (COMPACT)

Specifies whether device data compaction is performed. If the device specified does not support compaction, this parameter is ignored.

**\*FROMFILE**

Device data compaction is performed only if the file being read from the device specified on the **From device (FROMDEV)** parameter was written using device data compaction.

**\*YES** Device data compaction is performed on all files written to the device specified on the **To device (TODEV)** parameter.

**\*NO** Device data compaction is not performed.

Top

---

## Files to duplicate (FILES)

Specifies whether expired data files are copied from the tape volume placed in the device specified on the **From device (FROMDEV)** parameter to the tape volume placed on the device specified on the **To device (TODEV)** parameter.

**\*ALL** All data files on the tape volume are copied. All existing file sequence numbers are saved.

**\*ACTIVE**

Only data files with an expiration date later than the current system date are copied. Data files are renumbered consecutively, beginning with the number of the first file on the volume and omitting any files that have expired.

Top

---

## User label program (USRLBLPGM)

Specifies the name and library of the user program that processes user tape labels. For the device specified on the **To device (TODEV)** parameter, the user label program sends the user labels that are written to tape. For the device specified on the **From device (FROMDEV)** parameter, the user labels are sent to the user label program.

### Single values

**\*SYSCOPY**

User tape labels are processed to allow proper duplication of System/36 save and restore tapes. If user header labels are present on the tape volume specified on the FROMDEV parameter, they are copied to the tape volume specified on the TODEV parameter. The same is done for the user trailer labels at the end of the file or for the trailer labels at the end of the file section.

If an end-of-volume condition occurs on the device specified on the TODEV parameter before logical end-of-tape is found on the device specified on the FROMDEV parameter, user trailer and user header labels are created and written to the current and next tape volumes that replicate the data from the user header label read at the beginning of the file.

**\*NONE**

No user program processes user tape labels. No user labels are written to the tape volume.

### Qualifier 1: User label program

*name* Specify the name of the user program that processes the user tape labels.

### Qualifier 2: Library

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

**\*CURLIB**

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

*name* Specify the name of the library where the user label program is located.

Top

---

## From device end option (FROMENDOPT)

Specifies whether the tape volume placed on the device specified on the **From device (FROMDEV)** parameter is rewound, or rewound and unloaded after the operation is completed.

**\*REWIND**

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

**\*UNLOAD**

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

**\*LEAVE**

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

Top

---

## To device end option (TOENDOPT)

Specifies whether the tape volume placed on the device specified on the **To device (TODEV)** parameter is rewound, or rewound and unloaded after the operation is completed.

**\*UNLOAD**

The tape is rewound and unloaded after the operation is completed.

**\*REWIND**

The tape is rewound, but not unloaded.

**\*LEAVE**

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

Top

---

## Check for active files (CHECK)

Specifies whether a tape file on the volume mounted on the **To device (TODEV)** parameter is checked for active data before it is overwritten. If an unlabeled volume is on the **To device**, this parameter is ignored.

**\*YES** The file to be overwritten is checked for active data. Only the first file to be overwritten is checked for active data, any subsequent files are not checked. If active files are found, the operation is ended and an error message is sent.

**\*NO** Tape duplication continues with no checking for active files.

Top

---

## File expiration date (EXPDATE)

Specifies the expiration date to be assigned to all the files when they are copied. This parameter only applies to standard labeled tapes.

**\*FROMFILE**

The expiration date currently specified for the file to be copied is used.

**\*PERM**

All the copied files will be assigned a permanent expiration date.

*date* Specify the expiration date to be assigned to all the files when they are copied.

Top

---

## Examples

### Example 1: Duplicating a Single Volume to a Single Volume

```
DUPTAP FROMDEV(TAPE01) TODEV(TAPE02)
```

This command duplicates the tape volume mounted on device TAPE01 onto the tape volume mounted on device TAPE02.

### Example 2: Appending a Volume Set to the End of a Single Volume

```
DUPTAP FROMDEV(TAPE01) TODEV(TAPE02) FROMVOL( VOL001 VOL002)  
TOVOL(VOLABC) FROMSEQNBR(*ALL) TOSEQNBR(*END)
```

This command duplicates all files from the tape volumes VOL001 and VOL002 onto the end of the to-volume VOLABC on device TAPE02.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF516D

Data overwrite not allowed.

#### CPF67E6

Volume &2 is not correct

#### CPF67FA

Volume compatibility not correct.

#### CPF67FD

File sequence number &3 not correct for volume &2.

#### CPF67FE

No files found on volume &2.

#### CPF67F7

Continuation volume cannot be duplicated.

#### CPF67F8

TOSEQNBR not correct on volume &2.

#### CPF6708

Command ended due to error.

#### CPF671C

Not all files duplicated to device &2.

#### CPF6714

TOVOL(\*FROMVOL) not correct.

#### CPF6718

Cannot allocate device &1.

#### CPF6720

Incorrect volume &2 found on device &1.

#### CPF6721

Device &1 not a tape device.

#### CPF6722

End of tape found on device &1.

- CPF6725**  
Ending file sequence number less than starting sequence number.
- CPF6734**  
File sequence number &3 not found on volume &2.
- CPF6740**  
TODEV and FROMDEV must be different.
- CPF6751**  
Load failure occurred on device &4.
- CPF6754**  
Active file &4 found on volume &2.
- CPF676B**  
Volume on device &4 is write protected.
- CPF6760**  
Device &1 not ready.
- CPF6761**  
Labels not found while processing file.
- CPF6768**  
Volume on device &1 is write protected.
- CPF6772**  
Volume on device &1 cannot be processed.
- CPF9814**  
Device &1 not found.
- CPF9825**  
Not authorized to device &1.

[Top](#)



---

## Edit Authorization List (EDTAUTL)

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

Parameters  
Examples  
Error messages

The Edit Authorization List (EDTAUTL) command shows the list of users and their authorities. From this display, the user can add and remove users and change users' authorities on the authorization list.

**Restrictions:** You must have authorization list management (\*AUTLMGT) authority to, or ownership of, the specified authorization list.

Top

---

### Parameters

Keyword	Description	Choices	Notes
AUTL	Authorization list	<i>Qualifier list</i>	Required, Positional 1
	Qualifier 1: Authorization list	<i>Name</i>	

Top

---

### Authorization list (AUTL)

Specifies the authorization list to be edited.

This is a required parameter.

*name* Specify the name of the authorization list to be edited.

Top

---

### Examples

```
EDTAUTL  AUTL(MYLIST)
```

This command shows the authorization list MYLIST and allows it to be changed.

Top

---

### Error messages

#### \*ESCAPE Messages

##### CPF22B9

Not authorized to change authorities.

##### CPF2204

User profile &1 not found.

**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.

**CPF2211**  
Not able to allocate object &1 in &3 type \*&2.

**CPF2216**  
Not authorized to use library &1.

**CPF2217**  
Not authorized to user profile &1.

**CPF2283**  
Authorization list &1 does not exist.

**CPF9843**  
Object &1 in library &3 type &2 cannot be accessed.

[Top](#)

---

## Edit Backup List (EDTBCKUPL)

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

Parameters  
Examples  
Error messages

The Edit Backup List (EDTBCKUPL) command allows the user to select libraries and folders for backup. More information on backup is in the Backup and Recovery book, SC41-5304.

Top

---

### Parameters

Keyword	Description	Choices	Notes
BCKUPL	Backup list	*LIB, *FLR	Optional, Positional 1

Top

---

### Backup list (BCKUPL)

Specifies the backup list to be changed.

**\*LIB** The library backup list is changed.

**\*FLR** The folder backup list is changed.

Top

---

### Examples

```
EDTBCKUPL BCKUPL(*LIB)
```

This command displays the library backup list stored in user index QEZBACKUPL in library QUSRSYS, and allows the user to change it.

Top

---

### Error messages

#### \*ESCAPE Messages

##### CPF1EEA

Not authorized to library backup list.

##### CPF1E6B

Folder backup list in use.

##### CPF1E6D

Folder backup list damaged; new one created.

##### CPF1E65

Library backup list in use.

**CPF1E67**

Backup options and library backup list damaged.

**CPF1E99**

Unexpected error occurred.

**CPF7D41**

Error occurred while logging order assistance request.

**CPF7D42**

Error occurred while performing database operation.

**CPF9871**

Error occurred while processing.

[Top](#)

---

## Edit CHPND Constraints (EDTGPCST)

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

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

The Edit Check Pending Constraints (EDTGPCST) command shows a list of established constraints that have records that are possibly in violation of the constraints (check pending). From this display, you can verify and select or change the sequence of the constraints to be rebuilt during an initial program load (IPL).

This command is called while you are running an attended IPL if you have check pending constraints. From the display shown, you can select whether the system continues the IPL while verifying selected constraints, or continues the IPL after verifying selected constraints.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

#### Example 1: Editing a List of Constraints

EDTGPCST

This command shows you the referential constraints that are in check pending. You can edit the sequence for verifying the constraints from this display.

[Top](#)

---

### Error messages

#### \*ESCAPE Messages

CPF325C

Database object &1 is in error.

[Top](#)



---

## Edit DL File Attributes (EDTDLFA)

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

Parameters  
Examples  
Error messages

The Edit DataLink File Attributes (EDTDLFA) command allows the user to display and change the status of DataLinks.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

#### Editing a List of DataLinks

EDTDLFA

This command shows you the DataLinks that have pending DataLink requests. You can edit the sequence for attempting the pending linking or unlinking requests from this display.

[Top](#)

---

### Error messages

None

[Top](#)





---

## Edit DLO Authority (EDTDLOAUT)

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

Parameters  
Examples  
Error messages

The Edit Document Library Object Authority (EDTDLOAUT) command is used to change user authorization to a document or folder object.

The following information is displayed for the specified document or folder:

- The name of the document or folder
- The owner of the document or folder
- The name of the authorization list securing the document or folder (if there is one)
- Personal status of the document or folder
- A list of specific users authorized for the document or folder
- The authority given to the users with no specific authority (\*PUBLIC), who are not on the authorization list, and whose group has no authority for the document or folder

### Restrictions:

1. A user must have all (\*ALL) authority to the document or folder or all object (\*ALLOBJ) special authority to change the authority.
2. You must have authority to use the Add DLO Authority (ADDDLOAUT), Change DLO Authority (CHGDLOAUT) , and the Remove DLO Authority (RMVDLOAUT) commands to use this command.
3. The user must have \*ALLOBJ special authority to change the \*ROOT folder public authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Character value, *SYSOBJNAM, *ROOT	Required, Positional 1
FLR	Folder	Character value, *NONE	Optional
SYSOBJNAM	System object name	Name	Optional

Top

---

## Document library object (DLO)

Specifies the name of the document or folder for which user authority is being changed.

### \*SYSOBJNAM

User authority is being changed for the document or folder with the system object name specified on the **System object name (SYSOBJNAM)** parameter.

### \*ROOT

The public authority value of the \*ROOT folder is changed.

*character*

Specify the user-assigned name of the document or folder for which user authority is being changed. A maximum of 12 characters can be specified.

Top

---

## Folder (FLR)

Specifies the folder where the object specified for the **Document library object (DLO)** parameter is located.

\*NONE

A folder name is not specified. If a name is specified on the DLO parameter, and the object is located in a folder, \*NONE cannot be specified.

*character*

Specify the name of the folder that contains the object. The name can consist of a series of folder names if the folder containing the object is located in another folder. A maximum of 63 characters can be specified.

Top

---

## System object name (SYSOBJNAM)

Specifies the system object name of the folder or document. This parameter is valid only when \*SYSOBJNAM is specified on the **Document library object (DLO)** parameter.

*name* Specify the system object name for the folder or document for which user authority is being changed. You must specify 10 characters.

Top

---

## Examples

```
EDTDLOAUT DLO(DOCA) FLR(MYFLR)
```

This command allows the user of this command to change the list of authorized users and their authorities to the document library object named DOCA in folder MYFLR. The user of this command must have \*ALL authority to the object or be the owner of the object.

Top

---

## Error messages

\*ESCAPE Messages

**CPF8A77**

Folder &1 not found.

**CPF8A78**

Folder &1 in use.

**CPF8A79**

Folder &1 is logically damaged.

**CPF8A80**

Document &2 in use in folder &1.

- CPF8A82**  
Document &2 not found in folder &1.
- CPF8A88**  
Operation not allowed on document &2 in folder &1.
- CPF8A89**  
Document &2 in folder &1 is logically damaged.
- CPF89C0**  
You have specified an incorrect value.
- CPF90BA**  
Authority request for document library object failed.
- CPF90B6**  
You have specified an incorrect input value.
- CPF9073**  
No authority to view or change the security of document library object &1.
- CPF9079**  
Request to get document description not successful for user profile &1.
- CPF908A**  
Requester &1 not enrolled.
- CPF908B**  
Document library object not found.
- CPF909A**  
Document &2 in folder &1 is damaged.
- CPF9095**  
Folder &1 is damaged.
- 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



---

## Edit Document (EDTDOC)

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

Parameters  
Examples  
Error messages

The Edit Document (EDTDOC) command allows you to edit a document.

Top

---

### Parameters

Keyword	Description	Choices	Notes
DOC	Document	Character value, <u>*PRV</u>	Optional, Positional 1
FLR	Folder	Character value, <u>*PRV</u>	Optional, Positional 2
EXITPNL	Display exit panel	<u>*YES</u> , *NO	Optional, Positional 3

Top

---

### Document (DOC)

Specifies the name of the document to be edited.

\*PRV The name used in the previous session is used.

*document-name*

Specify the name of the document to be edited.

Top

---

### Folder (FLR)

Specifies the name of the folder that contains the document to be edited.

\*PRV The name used in the previous session is used.

*folder-name*

Specify the name of the folder that contains the document to be edited.

Top

---

### Display exit panel (EXITPNL)

Specifies whether the Exit Document display is shown when F3(Exit) or F12(Cancel) is pressed to end the editing.

- \*YES** The Exit Document display is shown when F3(Exit) or F12(Cancel) is pressed to end the editing.
- \*NO** The Exit Document display is not shown when F3(Exit) or F12(Cancel) is pressed to end the editing.

Top

---

## Examples

EDTDOC DOC(TASK4) FLR(INSTTXT)

This command displays the document TASK4 of the folder INSTTXT, and allows you to edit the document TASK4.

Top

---

## Error messages

### \*ESCAPE Messages

#### OFCFFFC

User storage capacity exceeded.

#### OFCFFFD

Damaged object found.

#### OFC8EA3

OfficeVision for AS/400 editor is not available to resolve to a display.

#### OFC80B5

OfficeVision for OS/400 editor is not available on the system.

#### OFC800A

Folder is in use.

#### OFC800B

Document &1 is in use.

#### OFC800F

Display does not support text.

#### OFC8006

Folder not found.

#### OFC8007

Document &1 not found in folder.

#### OFC8008

Request not allowed with folder.

#### OFC8009

Request not allowed with document &1.

#### OFC801A

Document has been saved to diskette, tape or save file.

#### OFC801D

Maximum number of text sessions active.

#### OFC801E

DW editor or text assist cannot be loaded.

- OFC8010**  
Document &1 cannot be processed.
- OFC8011**  
Document &1 needs to be recovered.
- OFC8016**  
Document &1 is checked out.
- OFC8018**  
Document &1 is empty.
- OFC8019**  
Required module not on system.
- OFC802E**  
Request failed for PC editor.
- OFC821B**  
Document &1 needs to be reclaimed.
- OFC8951**  
Data name must be specified.
- OFC8952**  
Type must be &9 or &10.
- OFC8953**  
Data &9 does not exist.
- OFC8954**  
Display terminal does not have graphics ability.
- OFC8955**  
PC Text-assist function required to view image.
- OFC903A**  
Document &1 is final form.
- OFC9811**  
Folder needs to be reclaimed.

[Top](#)





---

## Edit File (EDTF)

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

Parameters  
Examples  
Error messages

The Edit File (EDTF) command allows you to edit a stream file or a database file. This command can also be used to browse a file or directory.

Top

---

### Parameters

Keyword	Description	Choices	Notes
STMF	Stream file, or	<i>Path name</i>	Optional, Positional 1
FILE	Data base file	<i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Data base file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MBR	File member	<i>Name, *FIRST</i>	Optional

Top

---

### Stream file, or (STMF)

Specify the name of the file to be edited.

#### *stream-file-name*

Specifies the path name of the object or a pattern to match the name of the object to be edited.

The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern can be specified in the last part of the path name. An asterisk (\*) matches any number of characters. A list of all files or subdirectories that match the specified characters will be displayed. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes. If the file name specified is a directory, a list of the files and subdirectories is displayed. From this list you can edit or display the files in the directory.

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

Top

---

### Data base file (FILE)

Specifies the name of the database file to be edited.

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.

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

Top

---

## **File member (MBR)**

Specifies the name of the database file member to be edited.

The possible values are:

### **\*FIRST**

The first member in the database file is edited.

*member-name*

Specify the name of the member to be edited.

Top

---

## **Examples**

### **Example 1: Editing a Stream File**

```
EDTF STMF('/mydir/myfile.txt')
```

This command will start an edit session for file myfile.txt in directory mydir under the root directory.

### **Example 2: Editing a Database File Member**

```
EDTF FILE(MYLIB/MYFILE) MBR(MYMBR1)
```

This command will start an edit session for member MYMBR1 of file MYFILE in library MYLIB.

Top

---

## **Error messages**

### **\*ESCAPE Messages**

#### **CPFB601**

Display file QDZRUEDT missing or damaged.

#### **CPFB604**

Error processing line commands.

#### **CPFB605**

Invalid line command.

#### **CPFB609**

Cannot allocate work space.

#### **CPFB610**

Command not valid.

#### **CPFB611**

&1 occurrences of &3 changed. &2 not changed.

**CPF612**

Find or replacement string not specified.

**CPF613**

Error opening printer file.

**CPF614**

File name not specified.

**CPF615**

Target line not specified.

**CPF617**

CCSID not valid.

**CPF618**

Conversion between CCSID &1 and the job's CCSID is not supported.

**CPF619**

File is empty.

**CPF620**

&2

**CPF621**

File cannot be displayed and/or edited.

Top



---

## Edit DBCS Conversion Dict (EDTIGCDCT)

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

Parameters  
Examples  
Error messages

The Edit DBCS Conversion Dictionary (EDTIGCDCT) command lets you add, change, and delete alphanumeric entries and their related double-byte character set (DBCS) words from the specified DBCS conversion dictionary. The system refers to the DBCS conversion dictionary when performing DBCS conversion. The system displays the entries being edited when this command is specified.

**Note:** Use of the DBCS conversion function is not recommended for Chinese and Korean double-byte character sets.

Top

---

### Parameters

Keyword	Description	Choices	Notes
IGCDCT	DBCS conversion dictionary	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: DBCS conversion dictionary	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ENTRY	Dictionary entry	X'40'-X'FE', *ALL	Optional

Top

---

### DBCS conversion dictionary (IGCDCT)

Specifies the double-byte character set (DBCS) conversion dictionary to be edited and the library in which it is stored. If you do not specify a library name, the first dictionary found when searching the library list is edited.

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

*library-name*

Specify the library where the dictionary is located.

Top

---

### Dictionary entry (ENTRY)

Specifies the alphanumeric entries being edited with their related double-byte character set (DBCS) words.

**\*ALL** Any entry in the dictionary can be edited. The system first shows the Work with DBCS Conversion Dictionary display showing all alphanumeric entries in the dictionary. From this display, specific entries are chosen to be edited.

***generic\*-string***

Specify a character string of one or more characters followed by an asterisk (\*). All entries starting with the specified string can be edited. The system first shows the Work with DBCS Conversion Dictionary display. From this display, specific entries are chosen to be edited. If you do not include the asterisk, the system assumes that you want to edit a specific string.

The string cannot be longer than 12 characters.

***specific-string***

Specify a character string. The system shows the DBCS Edit Related Words display, showing a single alphanumeric entry and its related DBCS words. The specified entry can be edited.

The string cannot be longer than 12 characters. You can edit the related words on this display.

Top

---

## Examples

**Example 1: Showing the Work with DBCS Conversion Dictionary Display**

```
EDTIGCDCT  IGCDCT(DBCSLIB/QUSRIGCDCT)  ENTRY(123*)
```

This command shows the Work with DBCS Conversion Dictionary display showing all the alphanumeric entries that start with 123 in the dictionary QUSRIGCDCT, which is stored in the library DBCSLIB.

**Example 2: Showing the Edit Related Words Display**

```
EDTIGCDCT  IGCDCT(DBCSLIB/QUSRIGCDCT)  ENTRY(WORDS)
```

This command shows the Edit Related Words display showing the alphanumeric entry WORDS and its related words from the dictionary QUSRIGCDCT, which is stored in library DBCSLIB.

Top

---

## Error messages

**\*ESCAPE Messages**

**CPF2122**

Storage limit exceeded for user profile &1.

**CPF8138**

&8 damage on DBCS conversion dictionary &4 in &9.

**CPF8440**

Entries cannot be added to the system DBCS conversion dictionary.

**CPF8451**

Entry value &1 not correct.

**CPF8455**

Work station is not a DBCS device.

**CPF8461**

Entry &1 of DBCS conversion dictionary is logically damaged.

**CPF9801**

Object &2 in library &3 not found.

**CPF9802**

Not authorized to object &2 in &3.

**CPF9803**

Cannot allocate object &2 in library &3.

**CPF9810**

Library &1 not found.

**CPF9820**

Not authorized to use library &1.

**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





---

## Edit Library List (EDTLIBL)

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

Parameters  
Examples  
Error messages

The Edit Library List (EDTLIBL) command shows an entry display that allows you to make changes to the user portion of your library list. The Edit Library List display allows you to add libraries to the user portion of your library list, remove libraries from the user portion of your library list, and change the order of libraries in the user portion of your library list. This is an interactive command only.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

EDTLIBL

This command shows the Edit Library List display from which you can add libraries, remove libraries, and change the order of the libraries in the user portion of the library list.

[Top](#)

---

### Error messages

#### \*ESCAPE Messages

##### CPF2106

Library list not available.

##### CPF2184

Library list not replaced.

##### CPF2207

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

##### CPF2255

Command &1 failed the authority test.

[Top](#)



## Edit Object Authority (EDTOBJAUT)

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

Parameters  
Examples  
Error messages

The Edit Object Authority (EDTOBJAUT) command displays the list of authorized users of an object and their associated user authorities. If you own the object, or have \*ALLOBJ special authority, you can add, change or remove authority for the object. If you have object management authority for the object, you can remove your specific authorities or grant or remove them for other users.

The following are displayed for the specified object:

- The object name
- The name of the library containing the object
- The name of the object's owner
- The object's type
- A list of all the users who are authorized to use the object
- The authority that each user has for the object
- The authorization list name is displayed when the object is secured by an authorization list.

If an object does not have an owner name associated with it, no authorities for the object are shown.

### Restrictions:

1. The user must have object management authority to the object to use this command.
2. If the object is a file, the user must have object operational and object management authorities.
3. You must have \*USE authority to the auxiliary storage pool device if one is specified.

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, *CNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DTAARA, *DTADCT, *DTAQ, *EDTD, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCTBL, *IGCSRT, *IMGCLG, *IPXD, *JOB, *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, *PAGESEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDL, *PSFCFG, *QMFORM, *QMORY, *QRYDFN, *RCT, *S36, *SBS, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *TBL, *TIMZON, *USRIDX, *USRPRF, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 2
ASPDEV	ASP device	<i>Name, *, *SYSBAS</i>	Optional

---

## Object (OBJ)

Specifies the object for which the authorized users and their authorities are to be shown.

This is a required parameter.

*name* Specify the name of the object.

### Qualifier 2: Library

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

### \*CURLIB

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

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

---

## Object type (OBJTYPE)

The object type, such as command (\*CMD), file (\*FILE), or program (\*PGM), of the object whose authorized users and authorities are to be shown. To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt).

This is a required parameter.

---

## 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.

---

## Examples

```
EDTOBJAUT OBJ(ARLIB/PROG1) OBJTYPE(*PGM)
```

This command causes the list of authorized users and their authorities for the object named PROG1 to be shown, but only if the user has object management authority for the object. PROG1 is a program (\*PGM) located in the library named ARLIB.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF22B8

Not authorized to change authorities.

#### CPF22B9

Not authorized to change authorities.

#### CPF2204

User profile &1 not found.

#### 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.

#### CPF2211

Not able to allocate object &1 in &3 type \*&2.

#### CPF2216

Not authorized to use library &1.

#### CPF2217

Not authorized to user profile &1.

#### CPF2283

Authorization list &1 does not exist.

#### CPF9843

Object &1 in library &3 type &2 cannot be accessed.

Top



---

## Edit Questions and Answers (EDTQST)

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

Parameters  
Examples  
Error messages

The Edit Questions and Answers (EDTQST) command allows authorized users to edit questions and answers for publication in a specified database. More information is available in the Basic System Operation information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

### Restrictions:

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

Top

---

## Parameters

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

Top

---

## Q/A database (QSTDB)

Specifies the Q & A database in which to edit questions and answers.

The possible values are:

### \*SELECT

You are asked to specify a Q & A database. If only one Q & A database exists on the system, it is the default.

### *question-database*

Specify the name of the Q & A database in which to edit questions and answers.

Top

---

## Lib containing Q/A database (LIB)

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

The name of the Q & A database can be qualified by one of the following library values:

### **\*QSTLIB**

The library containing the specified Q & A database is searched. If \*SELECT is specified on the QSTDB parameter, any Q & A database in any library to which you are authorized can be selected.

#### ***library-name***

Specify the name of the library to be searched. If \*SELECT is specified on the QSTDB parameter, any Q & A database in the library to which you are authorized can be selected.

Top

---

## **Examples**

EDTQST

This command shows the Work with Candidate Questions display.

Top

---

## **Error messages**

None

Top



---

## Edit Rebuild of Access Paths (EDTRBDAP)

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

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

The Edit Rebuild of Access Paths (EDTRBDAP) command displays the Edit Rebuild of Access Paths menu, from which the access path information can be displayed or changed.

### Restrictions:

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

[Top](#)

---

## Parameters

None

[Top](#)

---

## Examples

EDTRBDAP

This command shows the controls that are available when editing rebuild access paths.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

**CPF325C**

Database object &1 is in error.

[Top](#)



---

## Edit Recovery for Access Paths (EDTRCYAP)

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

Parameters  
Examples  
Error messages

The Edit Recovery for Access Paths (EDTRCYAP) command shows a list of access path recovery times for the system and for auxiliary storage pools (ASP) that are currently active on the system. From this list, you can change target access path recovery times and view updated recovery status information. Additionally, the command will show up to 500 access paths with the largest estimated access path recovery time which are not eligible for system-managed access-path protection and why they are not eligible. Also, the command will show up to 500 access paths with the largest estimated access path recovery time which are currently being protected by system-managed access-path-protection.

The system uses no more than the specified amount of target access path recovery time when recovering access paths during an initial program load (IPL) or vary on of an independent ASP after an abnormal system end. Because the access path recovery time is a target, performance may range around the target.

The time taken to rebuild access paths exposed while running the Copy File (CPYF), the Reorganize Physical File Member (RGZPFM), or the Restore Object (RSTOBJ) commands is not considered in the target access path recovery time of access paths protected with this command.

You can use this command or the Change Recovery for Access Paths (CHGRCYAP) command to manage the protection of access paths that are not already protected through journaling.

For more information on using this command, see the "Journal management" topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

This command has no parameters.

### Restrictions:

- You must have job control (\*JOBCTL) special authority to use this command.
- This command is shipped with public \*EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use this command.
- If the current access path recovery state is \*OFF, the user must be in a restricted state to activate system-managed access-path protection by specifying a target access path recovery time value.
- If no user auxiliary storage pools (ASPs) exist on the system, an access path recovery time for ASP 1 cannot be specified. You must specify a system access path recovery time.

Top

---

## Parameters

None

Top

---

## Examples

EDTRCYAP

This command shows the Edit Recovery for Access Paths display from which you can show or modify the target access path recovery times for your system and configured user auxiliary storage pools (ASPs).

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF70FA

Recovery times reset before changes completed.

#### CPF70FB

No authority to use command.

#### CPF70FC

ASP time changes not valid with system time of \*OFF.

#### CPF70FE

ASP time changes not valid when system time is \*OFF.

#### CPF70F4

Error occurred.

#### CPF70F7

Restricted system required to change recovery times.

#### CPF70F9

Not all recovery time changes made active.

#### CPF700F

Access path recovery time for &1 set to \*NONE.

#### CPF701C

Change to system access path recovery time canceled.

#### CPF701D

Error occurred during change of recovery times.

#### CPF701E

Access path protection cannot be turned \*OFF.

#### CPF702E

Access path recovery times set to system defaults.

#### CPF9814

Device &1 not found.

#### CPF9825

Not authorized to device &1.

#### CPF9871

Error occurred while processing.

#### CPFB8ED

Device description &1 not correct for operation.

Top

---

## Edit S/36 Program Attributes (EDTS36PGMA)

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

Parameters  
Examples  
Error messages

The Edit System/36 Program Attributes (EDTS36PGMA) command presents the program attributes of the specified program on your display to allow you to change them. The attributes of a specified program or of all programs in the specified library can be changed. The attributes of a specified program, or of all programs in the specified library, can be changed.

Top

---

### Parameters

Keyword	Description	Choices	Notes
PGM	S/36 program	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: S/36 program	<i>Name</i> , *ALL	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	

Top

---

### S/36 program (PGM)

Specifies the name of the program having its attributes updated.

This is a required parameter.

**\*ALL** The attributes of all programs in the library are shown for update. \*ALL is not allowed if the library specified is \*LIBL.

*program-name*

Specify the name of the program.

The possible library values are:

**\*LIBL** The library list is used to locate the program.

**\*CURLIB**

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

*library-name*

Specify the library where the program is located.

Top

---

### Examples

```
EDTS36PGMA PGM(RPGLIB/*ALL)
```

This command shows the program attributes of all the programs in RPGLIB and allows them to be changed.

---

## Error messages

### \*ESCAPE Messages

**CPF2C01**

Program &1 attributes not changed.

**CPF2C02**

Changing attributes not allowed for SSP program &1.

**CPF2C03**

MRTMAX parameter value &3 not correct.

**CPF2C05**

Program name \*ALL not allowed with library \*LIBL.

**CPF7D41**

Error occurred while logging order assistance request.

**CPF7D42**

Error occurred while performing database operation.

**CPF9803**

Cannot allocate object &2 in library &3.

**CPF9811**

Program &1 in library &2 not found.

**CPF9820**

Not authorized to use library &1.

**CPF9830**

Cannot assign library &1.

**CPF9871**

Error occurred while processing.

---

## Edit S/36 Procedure Attributes (EDTS36PRCA)

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

Parameters  
Examples  
Error messages

The Edit System/36 Procedure Attributes (EDTS36PRCA) command presents the attributes of the specified procedure on your display for you to change. The attributes of a specified procedure, or of all procedures in the specified library, can be changed.

Top

---

### Parameters

Keyword	Description	Choices	Notes
<b>MBR</b>	S/36 procedure member	<i>Name</i> , *ALL	Required, Positional 1
<b>FILE</b>	Source file	<i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Source file	<i>Name</i> , <u>QS36PRC</u>	
	Qualifier 2: Library	<i>Name</i> , * <u>LIBL</u> , *CURLIB	

Top

---

### S/36 procedure member (MBR)

Specifies the name of the procedure member having its attributes updated.

This is a required parameter.

**\*ALL** The attributes of all procedure members in the file are shown for update.

*procedure-member-name*

Specify the name of the procedure member.

Top

---

### Source file (FILE)

Specifies the name of the physical file containing the procedure member.

QS36PRC

This is the name of the default physical file.

*source-file-name*

Specify the name of the physical file.

The possible library values are:

**\*LIBL** The library list is used to locate the file.

**\*CURLIB**

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

*library-name*

Specify the library where the file is located.

Top

---

## Examples

```
EDTS36PRCA MBR(RPGPROC) FILE(RPGLIB)
```

This command shows the attributes of procedure RPGPROC in file QS36PRC in library RPGLIB and allows them to be changed.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2C0A

Member &3 attributes not changed.

#### CPF2C0B

Changing attributes not allowed for SSP member &3.

#### CPF2C08

File &1 is not a source file.

#### CPF7D41

Error occurred while logging order assistance request.

#### CPF7D42

Error occurred while performing database operation.

#### CPF9803

Cannot allocate object &2 in library &3.

#### 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.

#### CPF9826

Cannot allocate file &2.

#### CPF9871

Error occurred while processing.

Top



---

## Edit S/36 Source Attributes (EDTS36SRCA)

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

Parameters  
Examples  
Error messages

The Edit System/36 Source Attributes (EDTS36SRCA) command presents the attributes of the specified source member on your display for you to change. The attributes of a specified source member, or of all source members in the specified library, can be changed.

Top

---

### Parameters

Keyword	Description	Choices	Notes
<b>MBR</b>	S/36 source member	<i>Name</i> , *ALL	Required, Positional 1
<b>FILE</b>	Source file	<i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Source file	<i>Name</i> , <u>QS36SRC</u>	
	Qualifier 2: Library	<i>Name</i> , * <u>LIBL</u> , *CURLIB	

Top

---

### S/36 source member (MBR)

Specifies the name of the source member that is having its attributes updated.

This is a required parameter.

**\*ALL** The attributes of all source members in the file are shown for update.

*source-member-name*

Specify the name of the source member.

Top

---

### Source file (FILE)

Specifies the name of the physical file that contains the source member.

QS36SRC

The default physical file, QS36SRC, is used.

*source-file-name*

Specify the name of the physical file.

The possible library values are:

**\*LIBL** The library list is used to locate the file.

**\*CURLIB**

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

*library-name*

Specify the library where the file is located.

Top

---

## Examples

```
EDTS36SRCA MBR(*ALL) FILE(SDALIB/QS36SRC)
```

This command shows the source attributes of all the source members in file QS36SRC in library SDALIB and allows them to be changed.

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPF2C0A**

Member &3 attributes not changed.

#### **CPF2C0B**

Changing attributes not allowed for SSP member &3.

#### **CPF2C08**

File &1 is not a source file.

#### **CPF7D41**

Error occurred while logging order assistance request.

#### **CPF7D42**

Error occurred while performing database operation.

#### **CPF9803**

Cannot allocate object &2 in library &3.

#### **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.

#### **CPF9826**

Cannot allocate file &2.

#### **CPF9871**

Error occurred while processing.

Top

---

## Edit Workstation Object Aut (EDTWSOAUT)

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

Parameters  
Examples  
Error messages

The Edit Workstation Object Authority (EDTWSOAUT) command shows a list of authorized users and their associated authorities to a specified workstation object. Workstation objects are used by the i5/OS Graphical Operations program. The owner of the object or the security officer can grant, change, or revoke authority to the object. If you have object management authority for the object, you can revoke your specific authorities, or you can grant them to or remove them from other users.

The following are shown for the specified object:

- The object name
- The name of the library containing the object
- The name of the object owner
- The object type
- A list of all the users who are authorized to use the object
- The authority that each user has for the object
- The authorization list name (if the object is secured by an authorization list)

If the object does not have an owner name associated with it, the authorities for the object are not shown.

### Restrictions:

1. The user must have object management authority to the object to use this command.
2. If the object is a file, the user must have object operational and object management authorities to use this command.

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, *TPLJOB, *JOB, *TPLJOBQ, *TPLJOBLOG, *JOBLOG, *TPLJOBQL, *JOBQL, *TPLMSGL, *MSGL, *TPLMSGQ, *TPLMSGSD, *MSGSD, *TPLSGNUSL, *SGNUSL, *TPLOBJL, *OBJL, *TPLLBSL, *LIBSL, *TPLLIB, *LAUNCH, *TPLLUNCH, *PRSET	

Top

---

## Workstation object type (WSOTYPE)

Specifies the workstation objects whose authority is to be edited.

This is a required parameter.

**\*TPLWRKARA**  
The work area template is the workstation object.

**\*WRKARA**  
The work area objects are the workstation objects.

**\*TPLPRTOL**  
The printer output list template is the workstation object.

**\*PRTOL**  
The printer output list objects are the workstation objects.

**\*TPLPRTL**  
The printer list template is the workstation object.

**\*PRTL** The printer list objects are the workstation objects.

**\*TPLOUTQ**  
The output queue template is the workstation object.

**\*TPLOUTQL**  
The output queue list template is the workstation object.

**\*OUTQL**  
The output queue list objects are the workstation objects.

**\*TPLJOBL**  
The job list template is the workstation object.

**\*JOBL** The job list objects are the workstation objects.

**\*TPLJOBQ**  
The job queue template is the workstation object.

**\*TPLJOBLOG**  
The job log template is the workstation object.

**\*JOBLOG**  
The job log objects are the workstation objects.

**\*TPLJOBQL**  
The job queue list template is the workstation object.

**\*JOBQL**  
The job queue list objects are the workstation objects.

**\*TPLMSGL**  
The message list template is the workstation object.

**\*MSGL**  
The message list objects are the workstation objects.

**\*TPLMSGQ**  
The message queue template is the workstation object.

**\*TPLMSGSEND**  
The message sender template is the workstation object.

**\*MSGSEND**  
The message sender objects are the workstation objects.

**\*TPLSGNUSL**  
The signed-on user list template is the workstation object.

**\*SGNUSL**  
The signed-on user list objects are the workstation objects.

**\*TPLOBJL**

The object list template is the workstation object.

**\*OBJL** The object list objects are the workstation objects.

**\*TPLLBSL**

The library list template is the workstation object.

**\*LIBSL**

The library list objects are the workstation objects.

**\*TPLLIB**

The library template is the workstation object.

**\*TPLLUNCH**

The job submitter template is the workstation object.

**\*LAUNCH**

The job submitter objects are the workstation objects.

**\*PRSSET**

The personal settings objects are the workstation objects.

Top

---

## Examples

```
EDTWSOAUT  WSOTYPE(*TPLMSGQ)
```

This command shows the list of authorized users to the message queue template.

Top

---

## Error messages

Unknown

Top



---

## Eject Emulation Output (EJTEMLOUT)

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

Parameters  
Examples  
Error messages

The Eject Emulation Output (EJTEMLOUT) command forces the last data received from the host system to the spooled file or printer by closing the printer file. After closing, another printer file is opened if \*IMMED was specified on the Start Printer Emulation (STRPRTEML) command, so that more data can be spooled or printed. Printing then starts if \*FILEEND was specified on the **Spooled output schedule (SCHEDULE)** parameter of the Create Printer File (CRTPRTF) command.

**Note:** When the emulation printer output is ejected, a page eject is also performed.

More information about device emulation is available in the 3270 Device Emulation Support book, SC41-5408.

### Additional Considerations

You must use care when running this command. Before entering the command, you should look at the printed output (if SPOOL(\*NO) was specified) or use the Display Spooled File (DSPSPLF) command to look at the spooled file (if SPOOL(\*YES) was specified), to determine whether the printer data is at a logical breaking point. If this function is requested when printer emulation is in the middle of a group of print data from the host system, the group is split into separate printer files on the system.

The effect of this command on the printer emulation output varies, depending on the values specified for the SPOOL and SCHEDULE parameters on the printer file.

The possible values and their conditions are:

- SPOOL(\*NO): All the data received from the host system is printed, and the printer moves to the top of the next page.
- SPOOL(\*YES) and SCHEDULE(\*IMMED): If a writer is active to the output queue and is printing this file, all the data received from the host system is printed, and the printer moves to the top of the next page. If a writer is not active (printing this file), the effect is the same as if SCHEDULE(\*FILEEND) was specified. Another printer file is opened on the output queue.
- SPOOL(\*YES) and SCHEDULE(\*FILEEND): The status of the printer file on the output queue changes from *open* to *ready to print*. If a writer is active, the data can be printed. Another printer file is opened on the output queue.
- SPOOL(\*YES) and SCHEDULE(\*JOBEND): The status of the printer file on the output queue changes from *open* to *closed*. The file is not ready to print until the end of the job is reached. Another printer file is opened on the output queue.

Top

---

## Parameters

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

Keyword	Description	Choices	Notes
PRTDEV	Print device	<i>Name</i>	Optional, Positional 3

Top

---

## Emulation device (EMLDEV)

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

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

Top

---

## Emulation location (EMLLOC)

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

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

Top

---

## Print device (PRTDEV)

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

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

Top

---

## Examples

```
EJTEMLOUT EMLDEV(HOSTPRT1)
```

This command closes the printer file in the printer emulation job using the emulation device HOSTPRT1, forcing the latest data from the host system out to the spooled file or printer.

Top



---

## Error messages

### \*ESCAPE Messages

#### CPF8595

Eject emulation output function not performed.

[Top](#)



---

## Element Definition (ELEM)

Parameters  
Examples  
Error messages

The Element (ELEM) command definition statements are used to define the elements of a mixed list parameter on a command. A list parameter is a parameter that accepts multiple values that are passed together as consecutive values pointed to by a single parameter. The values are preceded by a 2-byte binary value that indicates the number of elements defined for the parameter.

A list item is the value that represents one value among a group of values organized in a specific order in a list. If all of the list elements are not of the same type, one ELEM statement must be used for each element that appears in the list being defined. If all the elements are of the same type (a simple list), individual ELEM statements are not required. For a simple list, all that is necessary is to specify the number of elements in the list on the **Maximum values allowed (MAX)** parameter of the PARM statement.

The order in which the ELEM statements are entered into the source file determines their positional order in the list. The first ELEM statement (for the first list item) must have a statement label that matches the statement label on the **Type of value (TYPE)** parameter of the PARM or ELEM statements for the same list. The remaining ELEM statements in the list must be unlabeled. Lists of elements having different values can be nested to the depth of three levels, including the highest level. A maximum of 300 elements can be included in one list.

**Note:** The ELEM statement contains certain parameters and predefined values that can be used only when an IBM-supplied command processing program (CPP) is called by the command being defined. Because there are limitations in some high-level languages, these values may not be useful in the definition statements of user-defined commands. If the entire parameter is for IBM-supplied commands only, these parameters and values are identified by the phrase (*For IBM-supplied commands*) that immediately follows the parameter keyword or the predefined value to which it applies.

Top

---

### Parameters

Keyword	Description	Choices	Notes
TYPE	Type of value	<i>Simple name</i> , *DEC, *CHAR, *LGL, *NAME, *VARNAME, *INT2, *GENERIC, *INT4, *DATE, *TIME, *ZEROELEM, *HEX, *SNAME, *PNAME, *UINT2, *UINT4, *X, *CNAME	Required, Positional 1
LEN	Value length	Values (up to 3 repetitions): <i>Integer</i>	Optional, Positional 2
CONSTANT	Constant value	<i>Character value</i>	Optional
RSTD	Restricted values	*YES, <u>*NO</u>	Optional
DFT	Default value	<i>Character value</i>	Optional
VALUES	Valid values	Values (up to 300 repetitions): <i>Character value</i>	Optional
REL	Relational expression	<i>Element list</i>	Optional
	Element 1: Relational operator	*GT, *EQ, *GE, *NL, *LT, *NE, *LE, *NG	
	Element 2: Value	<i>Character value</i>	

Keyword	Description	Choices	Notes
RANGE	Range of values	<i>Element list</i>	Optional
	Element 1: Lower value	<i>Character value</i>	
	Element 2: Upper value	<i>Character value</i>	
SPCVAL	Special values	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: From value	<i>Character value</i>	
	Element 2: To replacement value	<i>Character value</i>	
SNGVAL	Single values	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: From value	<i>Character value</i>	
	Element 2: To replacement value	<i>Character value</i>	
MIN	Minimum values required	0-300, <u>0</u>	Optional
MAX	Maximum values allowed	<i>Integer</i> , <u>1</u>	Optional
ALWUNPRT	Allow unprintable characters	<u>*YES</u> , *NO	Optional
ALWVAR	Allow variable names	<u>*YES</u> , *NO	Optional
PGM	Is ELEM a program	<u>*NO</u> , *YES	Optional
DTAARA	Is ELEM a data area	<u>*NO</u> , *YES	Optional
FILE	If a file parameter, how used	<u>*NO</u> , *IN, *OUT, *UPD, *INOUT, *UNSPFD	Optional
FULL	Full field required	*YES, <u>*NO</u>	Optional
EXPR	Value an expression	<u>*NO</u> , *YES	Optional
VARY	Varying length	Single values: <u>*NO</u> Other values: <i>Element list</i>	Optional
	Element 1: Return length value	*YES	
	Element 2: Value length	<u>*INT2</u> , *INT4	
PASSATR	Pass attribute byte	*YES, <u>*NO</u>	Optional
CASE	Case of value	<u>*MONO</u> , *MIXED	Optional
CCSID	CCSID of value	<u>*JOB</u> , *UTF16	Optional
DSPINPUT	Display input	<u>*YES</u> , *PROMPT, *NO	Optional
CHOICE	Choice text	<i>Character value</i> , <u>*VALUES</u> , *NONE, *PGM	Optional
CHOICEPGM	Choice program	Single values: <u>*NONE</u> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Choice program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
INLPMTLEN	Initial prompt length	<u>*CALC</u> , *PWD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 17, 25, 32, 50, 80, 132, 256, 512	Optional
PROMPT	Prompt text or message ID	<i>Character value</i> , <u>*NONE</u>	Optional

Top

## Type of value (TYPE)

Specifies the type of list item being defined. The element can be an integer, a decimal or logical value, or a quoted or not quoted character string that can be a name, label, date, or time.

\*DEC The list item is a packed decimal number.

\*LGL The list item is a logical value, either a one ('1') or a zero ('0').

**\*CHAR**

The list item is a character string that can (optionally) be enclosed in apostrophes. If the character string contains any special characters (not including an asterisk (\*)), it *must* be enclosed in apostrophes. The maximum number of characters that can be in the character string is 5000.

**\*NAME**

The list item is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic or one of the special characters, \$, @, or #. The name can also be a string of characters starting and ending with double quotation marks (") or enclosed in parentheses. If a special value is used (as in \*LIBL or \*NONE), it should be specified on the **Special values (SPCVL)** parameter.

**\*SNAME**

The list item is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic or one of the special characters \$, @, or #. The remaining characters can be alphanumeric, an underscore, or one of the special characters \$, @, or #. The character string can be enclosed in parentheses. If a special value is used (as in \*LIBL or \*NONE), it must be specified on the **Special values (SPCVL)** parameter.

**\*CNAME**

The list item is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic or one of the special characters, \$, @, or #. If a special value is used (as in \*LIBL or \*NONE), it must be specified on the **Special values (SPCVL)** parameter.

**\*PNAME**

The list item is a character string that represents a path name string. Optionally the path name string may be enclosed in apostrophes. If the path name string contains any special characters (not including an asterisk (\*)), it must be enclosed in apostrophes. The maximum length of the path name string is 5000 characters.

**\*GENERIC**

The list item is a character string that represents a generic name. A generic name contains a maximum of 255 characters followed by an asterisk (\*) and must conform to the rules for generic names. The name identifies a group of objects whose names all begin with the characters preceding the asterisk. If an asterisk is not included, the system assumes that the generic name is a complete object name.

**\*DATE**

The list item is a character string that represents a date. When entering the command, the year may be specified with either 2 digits or 4 digits. If a 2-digit year is specified, the date is assumed to be in the range of January 1, 1940 through December 31, 2039. If a 4-digit year is specified, the date may be in the range of August 24, 1928 through May 9, 2071. When it is passed to the CPP, it is always passed in the format *Cyyymmdd*, where C = century, yy = year, mm = month, and dd = day. The century digit is set to 0 (zero) for years 19xx, and it is set to 1 (one) for years 20xx. When a date value is specified in this ELEM statement, it must be specified without quotation marks in one of the following formats: *mmddy*, *mmddyyyy*, or *Cyyymmdd*. If the user enters a date when the command is run, it must be specified in the job-date format. The job date separator may be used when the date is entered. If the separator character is used, the date must be enclosed in apostrophes.

**\*TIME**

The list item is a character string that represents a time. It is passed to the command processing program in a 6-byte character string as *hhmmss*, where hh = hours, mm = minutes, and ss = seconds. Values specified in this statement must be in the format *hhmmss*. When a user types a time in the command at run time, it must be specified in the format *hhmmss*. The job time separator may be used when the time is entered. If the separator character is used, the time must be enclosed in apostrophes.

**\*HEX** The list item value is hexadecimal in form. The specified characters must be 0 through F. They

are converted to hexadecimal (EBCDIC) characters (2 hex digits per byte), right-justified, and padded with zeros. If the value is enclosed in apostrophes, an even number of digits is required. If the value is not enclosed in apostrophes, the number of digits can be odd or even.

**\*ZEROELEM**

The list item is always considered as a list of zero elements, for which no value can be specified in the command. It is used to prevent a value from being entered for an element that is a list even though the CPP expects one. An element for which \*ZEROELEM is specified is not prompted for, although the other elements in the parameter are prompted and are passed to the CPP as a list.

**\*X** (For IBM-supplied commands) The list item value is a character string, variable name, or numeric value. The value is passed as a numeric value if it contains only digits, a + or - sign, or a decimal point; otherwise, it is passed as a character string.

**\*INT2** The list item is an integer that is passed as a 2-byte signed binary number.

**\*INT4** The list item is an integer that is passed as a 4-byte signed binary number.

**\*UINT2**

The list item is an integer that is passed as a 2-byte unsigned binary number.

**\*UINT4**

The list item is an integer that is passed as a 4-byte unsigned binary number.

**\*VARNAME**

(For IBM-supplied commands) The list item value is a CL variable name that is passed as a character string.

***statement-label***

The list item accepts a qualified list name or a mixed list of values. The statement label specified here by the TYPE parameter is the statement label that identifies the first of a series of QUAL or ELEM statements that further describe the qualified list name or the mixed list being defined. The label must be the same as the label specified by statement label on the **Type of value (TYPE)** parameter on the PARM statement for this list.

Top

---

## Value length (LEN)

Specifies the length of the list item value that is passed to the command processing program (CPP).

If \*INT2, \*INT4, \*UINT2, \*UINT4, \*DATE, \*TIME, \*ZEROELEM, or statement label is specified on the **Type of value (TYPE)** parameter, this parameter is not allowed.

If \*DEC is specified on the **Type of value (TYPE)** parameter, the decimal length is specified in the form (n1 n2), where n1 specifies the total number of digits in the value (including the decimal portion), and n2 specifies the number of allowable decimal digits to the right of the decimal point. (The value for n2 is optional. Zero is assumed if n2 is not entered.)

If \*X is specified for the **Type of value (TYPE)** parameter, the LEN parameter is used as follows:

- For character data, length1 specifies the minimum length to be passed. If a longer value is entered, the entire value is passed.
- For decimal data, length2 and length3 specify the length and decimal positions for a constant value. If a variable is entered, it is passed according to the variable attributes.
- For a logical value, length1 specifies the length of the value, which is always 1.

If TYPE is other than \*DEC or \*X, the second and third list elements must be omitted and simply the number of characters must be specified.

---

## Constant value (CONSTANT)

Specifies that a value is passed to the CPP as a constant for the list item when the command being defined is processed; the element is not to appear externally on the command. If specified, the value must satisfy the requirements specified by the following parameters:

- Type of value (TYPE)
- Value length (LEN)
- Valid values (VALUES)
- Relational expression (REL)
- Range of values (RANGE)
- Special values (SPCVAL)
- Full field required (FULL)

If a character constant is specified in this parameter, it can be no longer than 32 characters. This parameter is not valid in the following cases:

- If \*ZEROELEM is specified for the **Type of value (TYPE)** parameter.
- If \*YES is specified for the **Value an expression (EXPR)** parameter.
- If the **Maximum values allowed (MAX)** parameter is specified greater than 1.
- If a value is specified for the **Default value (DFT)** parameter.

If a constant is specified for the element being defined, no prompt text can be specified for the **Prompt text or message ID (PROMPT)** parameter. However, the other elements of the list parameter (of which this list item is a part) are still prompted, and their values along with this constant value are still passed to the CPP as a list.

Variables cannot be coded for this parameter.

---

## Restricted values (RSTD)

Specifies whether the value entered for the list item (specified in the ELEM statement) is restricted to only one of the values given in the **Valid values (VALUES)** parameter, the **Special values (SPCVAL)** parameter, or the **Single values (SNGVAL)** parameter; or if the value can be any value that satisfies the requirements specified by the following parameters:

- Type of value (TYPE parameter)
- Value length (LEN parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)
- Single values (SNGVAL parameter)
- Full field required (FULL parameter)

**\*NO** The value entered for the list item defined by this ELEM statement can be anything that matches the requirements specified by the following parameters:

- Type of value (TYPE parameter)
- Value length (LEN parameter)
- Relational expression (REL parameter)

- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)
- Single values (SNGVAL parameter)
- Full field required (FULL parameter)

**\*YES** The value entered for the list item in this ELEM statement is restricted to one of the values in the **Valid values (VALUES)** parameter, or to one of the from-values in the **Special values (SPCVAL)** parameter, or the **Single values (SNGVAL)** parameter. \*YES cannot be specified if a statement label or \*ZEROELEM is specified on the **Type of value (TYPE)** parameter.

Top

---

## Default value (DFT)

Specifies the default value that is assigned to the list item if the user does not specify a value. That is, the default value is used as the value of the list item if the user omits the parameter that represents this list item, or specifies \*N for the element, while coding or entering the command. The default value must satisfy one of the following:

- It must match the element requirements specified by the following parameters:
  - Type of value (TYPE parameter)
  - Value length (LEN parameter)
  - Relational expression (REL parameter)
  - Range of values (RANGE parameter)
  - Full field required (FULL parameter)
- It must be one of the from-values in the **Special values (SPCVAL)** parameter, or the **Single values (SNGVAL)** parameter.
- If the default is a character constant, it can have no more than 32 characters.
- If \*YES is specified on the **Restricted values (RSTD)** parameter, it must be in the list of values in the **Valid values (VALUES)** parameter, or in the list of from-values of the **Special values (SPCVAL)** parameter, or the **Single values (SNGVAL)** parameter.
- If this ELEM statement itself defines a list, the default value must be specified in the **Single values (SNGVAL)** parameter.

This parameter is valid only if the **Minimum values required (MIN)** parameter is 0, which means the element defined by this ELEM statement for this list is optional. This parameter is not allowed if the **Constant value (CONSTANT)** parameter is specified. A default cannot be specified if \*ZEROELEM is specified for the **Type of value (TYPE)** parameter; in that case, an assumed default is passed.

An *assumed* default value is not displayed by the command prompt; a blank input field is shown instead. If a default is specified in this parameter, it is displayed by the prompt exactly as specified.

*value* Specify the default value that meets the specified requirements or that is one of the values specified in the **Valid values (VALUES)** parameter, the **Special values (SPCVAL)** parameter, or the **Single values (SNGVAL)** parameter.

Variables cannot be coded for this value.

Top



---

## Valid values (VALUES)

Specifies a list of up to 300 constants (fixed values) from which one constant can be specified as the value of the list item. This parameter is valid only if all of the following are true:

- \*YES is specified for the **Restricted values (RSTD)** parameter.
- Both the **Range of values (RANGE)** parameter and the **Relational expression (REL)** parameter are *not* specified.
- Each constant matches the attributes specified by the following parameters:
  - Type of value (TYPE parameter)
  - Value length (LEN parameter)
  - Full field required (FULL parameter)

Character constants specified in this parameter can be no longer than 32 characters. Specify up to 300 constants that can be specified as the value of the list item. This parameter is not valid if a statement label or \*ZEROELEM is specified for the **Type of value (TYPE)** parameter;

If this ELEM statement is defining the first element in a list, the value specified for this parameter cannot be the same as the value specified in the **Single values (SNGVAL)** parameter on either the PARM or ELEM statement that points to this ELEM statement.

Top

---

## Relational expression (REL)

Specifies the relationship between the list item value and the value of another parameter or constant. The value associated with the referred to keyword is the value passed to the CPP, not the user-specified value. To specify the relationship, enter one of the following relational operators followed by a constant or the value of another parameter.

- \*LT less than
- \*LE less than or equal to
- \*EQ equal to
- \*GE greater than or equal to
- \*GT greater than
- \*NL not less than
- \*NE not equal to
- \*NG not greater than

This parameter is not valid if \*LGL, \*VARNAME, \*ZEROELEM, or a statement label is specified for the **Type of value (TYPE)** parameter, or if either the **Range of values (RANGE)** parameter or the **Valid values (VALUES)** parameter is specified. If \*CHAR (character type) is specified by the **Type of value (TYPE)** parameter, the EBCDIC value of the character string is used as an unsigned integer in the comparison. If a character constant is specified in this parameter, it can be no longer than 32 characters.

Top

---

## Range of values (RANGE)

Specifies the range, or the limits, for the value of the list item. The list item value must be greater than or equal to the lower limit value specified, and it must be less than or equal to the upper limit value specified. The value tested is the value sent to the CPP, not the user-specified value.

For nonnumeric data types, such as \*CHAR, the range of values and the data specified will be right-justified and padded on the left with blanks. A numeric range should not be used to define an interval for nonnumeric data unless leading zeros are specified or the data is only 1 character in length.

This parameter is not valid if either the **Relational expression (REL)** parameter or the **Valid values (VALUES)** parameter is specified, or if \*LGL, \*VARNAME, \*ZEROELEM, or statement label is specified on the **Type of value (TYPE)** parameter. Character constants specified in this parameter can be no longer than 32 characters.

Top

---

## Special values (SPCVL)

Specifies a list of up to 300 entries that define special values that can be entered for the element defined by this ELEM statement. Each entry specifies a character string (from value) that can be entered even though it may not meet all validity checking requirements. If the entered character string matches the from-value of one of the entries, and the to-value is specified, the string is replaced with the to-value and is then passed to the command processing program (CPP) without further checking. If the to-value is omitted, the from-value is passed to the CPP. This parameter is not valid if a statement label or \*ZEROELEM is specified for the **Type of value (TYPE)** parameter.

If a to-value of \*CURLIB is specified, the name of the current library is passed to the CPP rather than the value \*CURLIB. If the from-value is \*CURLIB and no to-value is specified, or if the to-value is \*CURLIB and it is enclosed in apostrophes, the value \*CURLIB is passed to the CPP.

The from-value is a character string, but the to-value can be anything that is passable. However, if \*DATE is specified for the **Type of value (TYPE)** parameter, the to-value must be specified not quoted in one of the following formats: *mmddy*, *mmddy*, or *cyymmdd*. If a CL variable is used for the from-value, its type must be \*CHAR. If this ELEM statement is defining the first element in a list, the value specified for the from-value cannot be the same as the value specified in the **Single values (SNGVAL)** parameter on either the PARM or ELEM statement that points to this ELEM statement.

The to-value must be no longer than is specified on the **Value length (LEN)** parameter; and, if \*DEC, \*INT2, \*INT4, \*UINT2 or \*UINT4 is specified for the **Type of value (TYPE)** parameter, the type of the to-value must be the same. If a character type (such as \*CHAR, \*LGL, or \*DATE) is specified for the **Type of value (TYPE)** parameter, the to-value must be a character string. Character constants specified in this parameter can be no longer than 32 characters. If a to-value is not specified, the from-value must be passable.

Variables cannot be coded for this element.

Top

---

## Single values (SNGVAL)

Specifies a list of up to 300 single values that can be specified for an element being defined as a statement label, or that is to have two or more list items in its nested list (defined by the **Maximum values allowed (MAX)** parameter). Any one of the single values can be used instead of a nested list of values or a qualified name that the element is defined to accept. Each entry specifies a character string

(from-value) that can be entered. If an entered character string matches the from-value of one of the entries and the to-value is specified, the data is replaced with the to-value and is then passed to the command processing program (CPP) without further checking. If the to-value is omitted, the from-value is passed to the CPP. If this ELEM statement is defining the first element in a list, the value specified for the from-value cannot be the same as the value specified in the **Single values (SNGVAL)** parameter on either the PARM or ELEM statement that points to this ELEM statement.

The to-value (or the from-value, if the to-value is omitted) must be passable, as specified in the **Special values (SPCVL)** parameter. Character constants specified in this parameter can be no longer than 32 characters. This parameter can be specified only if the **Maximum values allowed (MAX)** parameter is greater than one or if a statement label is specified for the **Type of value (TYPE)** parameter. It is not valid if \*ZEROELEM is specified for the **Type of value (TYPE)** parameter. Each single value can only substitute for a list of values or a qualified name; it cannot be a list item or qualifier. It is passed as the first element of the list.

If a to-value of \*CURLIB is specified, the name of the current library is passed to the CPP rather than the value \*CURLIB. If the from-value is \*CURLIB and no to-value is specified, or if the to-value is \*CURLIB and it is enclosed in apostrophes, the value \*CURLIB is passed to the CPP.

Variables cannot be coded for this element.

Top

---

## Minimum values required (MIN)

Specifies the minimum number of values that must be entered for the element being defined. For an element that does not allow multiple like values, only zero (0) for optional and one (1) for required can be specified as the minimum number of values.

For an element that allows multiple like values (because a value greater than one is specified in the **Maximum values allowed (MAX)** parameter), zero (0) indicates that no values must be entered; therefore, it is an *optional* element. A value of one (1) or greater than one indicates the minimum number of values that must be entered for the element, and therefore it is a *required* element. The value specified for this parameter cannot exceed the value specified for the **Maximum values allowed (MAX)** parameter. The number specified tells how many list items are required in another list. If this parameter is not specified, zero (0) is assumed, which means that the element is optional.

0      The list item is optional; it does not have to be entered.

### *minimum-number*

Specify the minimum number of elements that must be specified in the nested list. If 1 is assigned as the value, it specifies that one value is required for the element. If a number greater than 1 is specified, the element contains a list that must have at least as many elements as the number specified.

Top

---

## Maximum values allowed (MAX)

Specifies, if this ELEM statement is defining a simple list item, the maximum number of elements that this list item can have in its nested list. If a value greater than 1 is specified, the element is capable of accepting multiple like values (that is, a simple nested list). All values entered for this element (at the time the command is run) must satisfy the validity checking requirements specified by the other parameter values on this ELEM statement.

**Note:** The values for a nested list are passed consecutively, preceded by a 2-byte binary value that indicates the number of values entered in the list item by the user. CL programs do not support the handling of binary values in variables.

**1** The list item accepts only one value; there is no nested list.

#### *maximum-number*

Specify the maximum number of elements that the list item can accept. The specified maximum must be greater than or equal to the value specified in the **Minimum values required (MIN)** parameter and less than or equal to 300. If the maximum is greater than 1 and a statement label that identifies a QUAL statement or another ELEM statement is not specified for the **Type of value (TYPE)** parameter, the parameter, which is also an element, is a simple list of like values (that is, each element in the list has the same requirements, such as type and length). If a statement label is specified for the **Type of value (TYPE)** parameter, and it points to the label of a QUAL statement or another ELEM statement, this parameter should only be specified greater than 1 if a list of lists or a list of qualified names is to be accepted. A maximum greater than 1 is not valid if the **Constant value (CONSTANT)** parameter is also specified.

Top

---

## Allow unprintable characters (ALWUNPRT)

Specifies whether this ELEM statement should accept the hexadecimal characters above X'FF' and those in the range of X'00' to X'3F'. This parameter is valid only if \*CHAR or \*X is specified for the **Type of value (TYPE)** parameter.

**\*YES** Any characters can be sent to the display or printer.

**\*NO** Unprintable characters cannot be passed to the command processing program.

Top

---

## Allow variable names (ALWVAR)

Specifies whether variable names are allowed for the element. \*NO is not allowed if \*VARNAME, \*ZEROELEM, \*NULL, or a statement label is specified for the **Type of value (TYPE)** parameter.

**\*YES** Variable names can be used for the element.

**\*NO** Variable names cannot be used for the element.

Top

---

## Is ELEM a program (PGM)

Specifies whether this element is a program name. \*YES is valid only if a statement label, \*CHAR, \*NAME, \*SNAME, \*CNAME, or \*GENERIC is specified for the **Type of value (TYPE)** parameter. The specification of \*YES on this parameter does not have any effect on the element being defined by the ELEM statement; it only indicates to the compiler that the value for this element is a program name. This information is stored so that it can be included in the output of the Display Program References (DSPPGMREF) command.

**\*NO** The element (defined in this ELEM statement) is not a program name.

**\*YES** The element is a program name.

Top

---

## Is ELEM a data area (DTAARA)

Specifies whether the element is a data area name. \*YES is valid only if a statement label, \*CHAR, \*NAME, \*SNAME, \*CNAME, or \*GENERIC is specified for the **Type of value (TYPE)** parameter. The specification of \*YES on this parameter does not have any effect on the element being defined by the ELEM statement; it only indicates to the compiler that the value for this element is a data area. This information is stored so that it can be included in the output of the Display Program References (DSPPGMREF) command.

**\*NO** The element (defined in this ELEM statement) is not a data area name.

**\*YES** The element is a data area name.

Top

---

## If a file parameter, how used (FILE)

Specifies whether if the list item is a file name and the expected use of the file. The element can be specified as the name of a file that has a specific use so that, at compile time, the names can be used to get file reference information about where the files are used. This parameter is valid only if a statement label, \*CHAR, \*NAME, \*SNAME, \*CNAME, or \*GENERIC value is specified for the **Type of value (TYPE)** parameter. The specification in this parameter does not have any effect on the list item being defined by the ELEM statement; it only indicates to the compiler that the value for this element is a file name and what type of file it is. This information is stored so that it can be included in the output of the DSPPGMREF (Display Program References) command. One of the following types of files can be specified:

**\*NO** The list item (defined in this ELEM statement) is not a file name.

**\*IN** The list item is an input file name.

**\*OUT** The list item is an output file name.

**\*UPD** The list item is an update file name.

**\*INOUT**

The list item value is the name of a file that is to be used for both input and output.

**\*UNSPFD**

The list item value is the name of a file, but its use cannot be specified.

Top

---

## Full field required (FULL)

Specifies whether the number of characters in the list item must be exactly the same as the number specified in the **Value length (LEN)** parameter (if specified) or its default length (if LEN is not specified).

**\*NO** The number of characters in the list item can be less than that specified by the **Value length (LEN)** parameter.

**\*YES** The number of characters in the list item must equal the number specified by the **Value length (LEN)** parameter, or the default length for that type. The exact length is valid only for the following types on the **Type of value (TYPE)** parameter: \*LGL, \*CHAR, \*NAME, \*SNAME, \*CNAME, \*GENERIC, \*VARNAME, and \*HEX.

Top

---

## Value an expression (EXPR)

Specifies whether the element can accept an expression containing a character concatenation.

**\*NO** The element value cannot be a concatenation expression.

**\*YES** The element value can be a concatenation expression. \*YES is not valid if a value is specified for the **Constant value (CONSTANT)** parameter.

Top

---

## Varying length (VARY)

Specifies whether the list item value that is passed to the CPP is preceded by a length value that indicates the number of characters entered for the element's value.

### Single values

**\*NO** The element value is not preceded by a length value.

### Element 1: Return length value

**\*YES** The element value passed to the CPP is preceded by a field that indicates the number of characters actually specified for the parameter. \*YES is valid only for the following parameter types: \*CHAR, \*NAME, \*SNAME, \*CNAME, \*PNAME, \*GENERIC, \*LGL, and \*VARNAME. \*YES must be specified if PASSATR(\*YES) and RTNVAL(\*YES) are specified.

**Note:** The length value is the actual number of characters entered for the command parameter with trailing blanks removed. The length value passed may be different than the defined parameter length or the declared variable length. The length of the field containing the character string data is determined by the defined length for the parameter or the declared LEN for CL Program variables. The length value defines how many characters in the character string data field were actually entered for the command parameter.

### Element 2: Value length

**\*INT2** The element value is an integer passed as a 2-byte signed binary number.

**\*INT4** The element value is an integer passed as a 4-byte signed binary number.

Top

---

## Pass attribute byte (PASSATR)

Specifies whether an attribute byte is to be passed to the command processing program (CPP) with the list item data. This parameter is not valid if a statement label or \*ZEROELEM is specified for the **Type of value (TYPE)** parameter.

The attribute byte precedes the list item data. If the list item allows multiple values to be specified, an attribute byte precedes each value.

**\*NO** No attribute byte is passed with the list item.

**\*YES** An attribute byte is passed with the list item.

The attribute byte has two fields:

1. The leftmost bit of the attribute byte indicates whether or not a value was specified. If the leftmost bit is '0'B, the value passed to the command processing program is a default value and was not specified in the command string. If the leftmost bit is '1'B, the value passed to the command processing program was specified in the command string.

2. The remaining seven bits describe the value passed to the command processing program when \*CHAR is specified for the **Type of value (TYPE)** parameter.

Attribute	Description
'0000010'B	Meets *NAME rules, like <b>A_B</b>
'0000100'B	Meets *GENERIC rules, like <b>AB*</b>
'1000101'B	Quoted character string, like <b>'A B'</b>
'0000101'B	Unquoted character string, like <b>5A</b>
'1001000'B	Logical constant, <b>'0'</b> or <b>'1'</b>
'0001100'B	Hexadecimal value, like <b>X'C1C2'</b>
'0100001'B	Unsigned numeric value, like <b>5</b>
'0101001'B	Unsigned numeric with decimal point, like <b>5.2</b>
'0110001'B	Signed numeric value, like <b>-5</b>
'0111001'B	Signed numeric with decimal point, like <b>-5.2</b>

Top

---

## Case of value (CASE)

Specifies whether the value that is passed to the CPP is changed from lowercase to uppercase, or is preserved in the case specified on for the command parameter.

### \*MONO

The element value is changed from lowercase to uppercase. Parameters enclosed with apostrophes preserve the case whether or not this value is specified.

### \*MIXED

The element value is preserved in the case specified on the command parameter. The value can be specified only for \*CHAR and \*PNAME parameter types.

Top

---

## CCSID of value (CCSID)

Specifies the coded character set identifier (CCSID) to use when passing the element value.

\*JOB If the command string was originally in Unicode, the value will be converted to the job CCSID. If the original command string was not in Unicode, the job CCSID is assumed and no conversion is done.

### \*UTF16

The element value is converted to UTF16. If the original input was not in Unicode it is assumed to be in the job CCSID. The value can be specified only for \*CHAR and \*PNAME element types.

Top

---

## Display input (DSPINPUT)

Identifies if the keyword value is to be shown in the job log or in a prompt display.

\*YES The default response, \*YES, indicates that the parameter value will be shown on the prompt display and in the job log.

### \*PROMPT

The response \*PROMPT indicates that the parameter value will be shown on the prompt display but not in the job log.



**\*NO** The response \*NO indicates that the parameter values will not be shown on either the prompt display or in the job log.

Top

---

## Choice text (CHOICE)

Specifies the choices text that is displayed to the right of the input field on the prompt screen. Up to 30 characters of text can be displayed.

### **\*VALUES**

The choices text is generated based on the values specified for the TYPE, RSTD, RANGE, SNGVAL, SPCVAL, and VALUES parameters. If constants are specified for the RANGE parameter, the choices text begins with the minimum value and the maximum value separated by a hyphen. If RANGE is not specified with constants as the minimum and maximum values, and RSTD(\*NO) is specified, the choices text begins with a short description of the parameter type based on the value specified for the TYPE parameter. Values specified for the SNGVAL parameter are added to the choices text, in the order the values are defined in the command definition source and separated by a comma and a blank. The last entries added to the choices text are values specified for the SPCVAL or VALUES parameter, in the order the values are defined in the command definition source and separated by a comma and a blank. If there are too many values to fit in 30 characters, the last value is followed by three periods.

The following are examples of possible choices text generated by CHOICE(\*VALUES):

- If TYPE(\*DEC) and RANGE(1.0 999.9) and SPCVAL((\*NOMAX -1)) are specified, the choices text will be:  
1.0-999.9, \*NOMAX
- If TYPE(\*NAME) and RSTD(\*NO) and SNGVAL(\*ALL) and SPCVAL(\*LIBL \*CURLIB) are specified, the choices text will be:  
Name, \*ALL, \*LIBL, \*CURLIB
- If RSTD(\*YES) and SNGVAL(\*ALL) and SPCVAL(\*ALRTBL \*BNDDIR \*CHTFMT \*CLD \*CLS \*CMD) are specified, the choices text will be:  
\*ALL, \*ALRTBL, \*BNDDIR...

### **\*NONE**

No values are displayed.

**\*PGM** A program that is called determines the values that are displayed. The program that is called is identified in **Choice program (CHOICEPGM)** parameter of the PARM statement.

### *message-identifier*

Specify the message ID of the message used to retrieve the message containing the text for the possible values field. The message file specified on the **Message file for prompt text (PMTFILE)** parameter of the Create Command (CRTCMD) command is used to find the message.

### *'choices-text'*

Specify no more than 30 characters, enclosed in apostrophes.

Top

---

## Choice program (CHOICEPGM)

Specifies the program to be called during command prompting to fill in the possible choices text and the permissible values. This parameter must be specified if \*PGM is specified on the **Choice text (CHOICE)** parameter and may not be specified otherwise.

### Single values



### \*NONE

No program is identified to fill in the possible choices text and permissible values.

### **Qualifier 1: Choice program**

*name* Specifies the name of the program to be called during prompting to fill in the possible choices text or permissible values. If an exception occurs when the program is called, no possible choices text is left blank, and the list of permissible values is taken from the command.

### **Qualifier 2: Library**

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

### \*CURLIB

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

*name* Specify the name of the library where the program is located.

Top

---

## **Initial prompt length (INLPMTLEN)**

Specifies the length of the input field initially displayed for the element when the command is prompted. The user can extend the field to a maximum length of 512 bytes by entering an ampersand (&) in the first position of the field, followed by a blank. INLPMTLEN is valid only if TYPE is specified as \*CHAR, \*NAME, \*SNAME, \*CNAME, \*PNAME, \*GENERIC, or \*HEX. If FULL(\*YES), RSTD(\*YES), or CONSTANT are specified, INLPMTLEN(\*CALC) must be specified or defaulted.

### \*CALC

The prompter will determine the length of the prompt field based on the type and length of the parameter.

**\*PWD** If the current value of system value QPWDLVL is '0' or '1', the prompt field will be 10 bytes long. Otherwise, the length of the prompt field will be determined by the length of the parameter. INLPMTLEN(\*PWD) is valid only if TYPE is specified as \*CHAR, \*NAME, \*SNAME, \*PNAME, or \*CNAME.

### *initial-prompt-length*

Specify the initial length in bytes. Valid values are 1-12, 17, 25, 32, 50, 80, 132, 256, and 512.

Top

---

## **Prompt text or message ID (PROMPT)**

Specifies the prompt text, if any, is that used for the list item (defined in this ELEM statement). The prompt text gives a short description of the element which appears next to the element input field when the command is prompted. Prompt text cannot be specified if \*ZEROELEM is specified for the **Type of value (TYPE)** parameter, or if a constant value is specified for the **Constant value (CONSTANT)** parameter.

### \*NONE

No prompt text is displayed for the list item defined by this ELEM statement. This list item is still prompted by an input field, but no text is displayed with it.

### *message-identifier*

Specify the message identifier that specifies the message containing the prompt text of up to 30 characters that is displayed when the program is prompting for the list item. If a message having

the specified identifier cannot be found in the message file specified in the **Message file for prompt text (PMTFILE)** parameter of the Create Command (CRTCMD) command, the message identifier itself is used as the prompt text.

*'prompt-text'*

Specify the prompt text that is displayed when the program is prompting for the list item. The text must be a character string of no more than 30 characters, enclosed in apostrophes.

Top

---

## Examples

### Example 1: Define a Parameter with Two Different Types of Elements

```
      PARM  KWD(JOBDESC)  TYPE(L1)  MIN(1)
L1:  ELEM  TYPE(*NAME)  LEN(10)  MIN(1)
      ELEM  TYPE(*DEC)  LEN(2)  MIN(1)  REL(*LE 60)
```

The parameter named JOBDESC is required and has two elements which must both be specified. The first element is a ten-character name, and the second element is a 2-digit number that is less than or equal to 60.

### Example 2: Define a Parameter with Similar Elements

```
      PARM  KWD(RANGE)  TYPE(L1)  MIN( DFT(*SAME) +
      SNGVAL((*SAME 101))
L1:  ELEM  TYPE(*DEC)  MIN(1)  REL(*LE 100)
      ELEM  TYPE(*DEC)  MIN(1)  REL(*LE 100)
```

The parameter named RANGE can be omitted, but, if present, it must be a list of two numbers, neither of which can be greater than 100. To allow the command processing program to determine whether the value passed is a user-specified value or the \*SAME single value, \*SAME is mapped to 101 which is outside the normal range of values being checked for.

Top

---

## Error messages

None

Top

---

## Else (ELSE)

### Where allowed to run:

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

**Threadsafe:** Yes

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

The Else (ELSE) command is used with an IF command to specify another command that is to be conditionally processed. The ELSE command is processed only if the result of evaluating the logical expression on the preceding IF command is false. If the result is true, the ELSE command and commands associated with it are not processed.

The ELSE command can specify a CL command, or a Do group, to be processed for the false condition.

An ELSE command does not have to follow each IF command, but each ELSE command that is coded must have an associated IF command preceding it. If nested levels of IF commands are used, a given ELSE is always matched with the innermost IF command that has not already been matched with another ELSE command. Although the ELSE command is optional, coding all of the matching ELSE commands makes it easier to see where all of the nesting levels start and end.

**Restrictions:** The ELSE command is valid only in a CL procedure. It must have an associated IF command preceding it.

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
CMD	Command	<i>Command string</i>	Optional, Positional 1

[Top](#)

---

## Command (CMD)

Specifies the command or commands (in a Do group) to be processed if the result of evaluating the expression on the corresponding IF command is false.

If the command specified in this parameter is a DO command, all of the commands specified within the Do group are considered to be part of the command specified by the parameter. If no command is specified, no action is taken for a false condition.

If the command specified by the CMD keyword is not coded on the same line as the keyword, the left parenthesis following CMD must be coded on the same line, followed by a + or - to show continuation. The command and the right parenthesis can then be coded on the next line. For example:

```
ELSE CMD( +  
      GOTO C)
```

If any part of the command continues on the next line, a continuation character (+ or -) must be specified.

If a DO command is specified, only the DO command (not the commands specified as part of the Do group) is placed in parentheses. For example:

```
ELSE CMD(DO)
  CMD1
  CMD2
  .
  .
  .
ENDDO
```

The following commands, although valid in CL procedures, cannot be specified on the ELSE command:

- ENDDO (End Do)
- MONMSG (Monitor Message)
- PGM (Program)
- ENDPGM (End Program)
- DCL (Declare CL Variable)
- DCLF (Declare File)
- another ELSE command
- WHEN, OTHERWISE, ENDSELECT

In addition, the MONMSG command cannot be specified as the next command after the ELSE command.

Top

---

## Examples

### Example 1: Using ELSE and IF Commands

```
IF (&A *GT &B) THEN(CHGVAR VAR(&A) VALUE(&B))
ELSE (CHGVAR &B &A)
```

If the value of &A is greater than the value of &B, &A is set equal to &B. If &A is less than or equal to &B, the test result is false. The CHGVAR command on the ELSE command is processed, and the value of &B is set to the same value as &A. (Refer to the CHGVAR (Change Variable) command for the description of the command and its parameters.)

### Example 2: Nested Levels of Commands

```
IF COND(&A *EQ &B) +
THEN(IF (&C *EQ &D) +
THEN(IF (&C *EQ &F) THEN(DO)))
CMD1
CMD2
.
.
ENDDO
ELSE CMDX
ELSE CMDY
ELSE DO
```

This example shows the use of nested levels of IF commands where an ELSE command is associated with each IF. The use of the ELSE commands makes the nested levels of IF commands easier to identify.

Top

---

## Error messages

None





---

## Emulate Printer Keys (EMLPRTKEY)

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

Parameters  
Examples  
Error messages

The Emulate Printer Key (EMLPRTKEY) command causes the printer emulation job or session that is using the specified printer emulation device to send either a PA1 or PA2 key signal to the host system.

PA keys are program access keys that are used to signal the host system. The host system program determines how these keys work. This command can only be used on 3287 or 3289 emulated printers operating as an LU type 1 session. In addition, the PA key signal, although sent to the host system, may not immediately be received.

Top

---

### Parameters

Keyword	Description	Choices	Notes
EMLDEV	Emulation device, or	<i>Name</i>	Optional, Positional 1
EMLLOC	Emulation location	<i>Communications name</i>	Optional, Positional 2
PRTDEV	Print device	<i>Name</i>	Optional, Positional 3
PRTKEY	Emulated printer key	*PA1, *PA2	Optional

Top

---

### Emulation device (EMLDEV)

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

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

Top

---

## Emulation location (EMLLOC)

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

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

[Top](#)

---

## Print device (PRTDEV)

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

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

[Top](#)

---

## Emulated printer key (PRTKEY)

Specifies the PA key signal that is sent to the host system. The host system program determines how these keys work.

**\*PA1** The PA1 key signal is sent to the host system.

**\*PA2** The PA2 key signal is sent to the host system.

[Top](#)

---

## Examples

```
EMLPRTKEY EMLDEV(HOSTPRT2) PRTKEY(*PA2)
```

This command causes the printer emulation session using emulation device HOSTPRT2 to send the PA2 key signal to the host system.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF8598

Emulate print key function not performed.

[Top](#)



---

## End Agent Services (ENDAGTSRV)

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

Parameters  
Examples  
Error messages

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

You can restart the agent services by running the Start Agent Services (STRAGTSRV) command.

### Restrictions:

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

Top

---

## Parameters

Keyword	Description	Choices	Notes
PREFDIR	Preferences file directory	<i>Path name</i> , *DFT	Optional

Top

---

## Preferences file directory (PREFDIR)

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

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

### *path-name*

Specify the directory that contains the preferences files to be used to end the agent services.

Top

---

## Examples

### Example 1: Ending with Shipped Default Values

```
ENDAGTSRV
```

This command ends all of the agent services defined in the default *ableplatform.preferences* file in '/qibm/prodData/OS400/able/' in a way defined in the default *able.preferences* file in '/qibm/prodData/OS400/able/'.

### Example 2: Ending with User-Specified Values

```
ENDAGTSRV  PREFDIR('/qibm/userData/OS400/able/test/')
```

This command ends all of the running agent services defined in the ableplatform.preferences file in '/qibm/userData/OS400/able/test/' in a way defined in the default able.preferences file in '/qibm/userData/OS400/able/test/'.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1890

\*ALLOBJ authority required for requested operation.

#### CPF90FF

\*JOBCTL special authority required to do requested operation.

### Error messages from submitted job:

This command submits a batch job which will end the batch jobs where the agent services are running. The following error messages could be signaled from this batch job:

#### CPF4B03

Java Virtual Machine(JVM) exception has occurred.

#### CPF4B04

Unable to finish ending agent services. Reason code &1

Top

---

## End ASP Balance (ENDASPBAL)

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

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

The End ASP Balance (ENDASPBAL) command allows the user to end the ASP balance function that was started using the Start ASP Balance (STRASPBAL) CL command. A message will be sent to the system history (QHST) log when the ASP balance function is ended for each ASP.

For more information about ASP balancing, see the Hierarchical Storage Management Use, SC41-5351.

### Restrictions:

- You must have all object (\*ALLOBJ) special authority to run this command.

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
ASP	ASP number	Single values: *ALL Other values (up to 32 repetitions): 1-32	Optional, Positional 1
ASPDEV	ASP device	Single values: *ALLAVL Other values (up to 32 repetitions): <i>Name</i>	Optional

[Top](#)

---

## ASP number (ASP)

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

**Note:** A value must be specified for either the **ASP number (ASP)** parameter or the **ASP device (ASPDEV)** parameter. Both parameters may be specified.

### Single values

**\*ALL** ASP balancing will be ended for the system ASP (ASP number 1) and all basic ASPs (ASP numbers 2-32) defined to the system.

### Other values (up to 32 repetitions)

**1-32** Specify the number of the ASP for which ASP balancing is to be ended.

[Top](#)

---

## ASP device (ASPDEV)

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

**Note:** A value must be specified for either the **ASP number (ASP)** parameter or the **ASP device (ASPDEV)** parameter. Both parameters may be specified.

### Single values

#### **\*ALLAVL**

ASP balancing will end for all ASP devices that currently have a status of 'Available'.

### Other values (up to 32 repetitions)

*name* Specify the name of the independent ASP device for which ASP balancing is to be ended.

Top

---

## Examples

### Example 1: End ASP Balance for ASP 1

```
ENDASPBAL ASP(1)
```

This command allows the user to end the ASP balancing function for ASP 1.

### Example 2: End ASP Balance for ASPs 1-32

```
ENDASPBAL ASP(*ALL)
```

This command will end the ASP balancing functions for the system ASP (ASP number 1) and each basic ASP (ASP numbers 2-32) that is currently being balanced.

### Example 3: End ASP Balance for an ASP Device

```
ENDASPBAL ASPDEV(MYASP1)
```

This command will end the ASP balancing function for ASP device MYASP1.

### Example 4: End ASP Balancing for All ASPs

```
ENDASPBAL ASP(*ALL) ASPDEV(*ALLAVL)
```

This command will end the ASP balancing functions that may be active on ASP numbers 1-32 and all ASP devices that have a status of 'Available'.

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPF18AC**

ASP balancing not active for ASP &1.

#### **CPF18AD**

ASP &1 must contain more than a single unit.

#### **CPF1890**

\*ALLOBJ authority required for requested operation.

#### **CPF9829**

Auxiliary storage pool &1 not found.

Top

---

## End Batch Job (ENDBCHJOB)

### Where allowed to run:

- Batch job (\*BATCH)

**Threadsafe:** No

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

The End Batch Job (ENDBCHJOB) command is a delimiter in a batch input stream that indicates the end of a job. The End Batch Job (ENDBCHJOB) command also can indicate the end of an inline data file provided the command is detected while the inline file is being processed.

**Restrictions:** This command cannot be entered at a work station. The End Batch Job (ENDBCHJOB) command must be preceded by two slashes (//) in positions 1 and 2 of the data record, for example //ENDBCHJOB. Blanks can separate the slashes from the command name (// ENDBCHJOB).

There are no parameters for this command.

[Top](#)

---

## Parameters

None

[Top](#)

---

## Examples

```
//ENDBCHJOB
```

This command indicates the end of a job that began with the BCHJOB command.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF1753

Command cannot be run.

[Top](#)



---

## End CHT Server (ENDCHTSVR)

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

Parameters  
Examples  
Error messages

The End Clustered Hash Table Server (ENDCHTSVR) command is used to end the specified clustered hash table server on the cluster nodes specified by the NODE parameter. This will remove the specified nodes from the clustered hash table domain. If all of the cluster nodes defined in the clustered hash table domain are specified on the NODE parameter the server job will be ended on all cluster nodes and the clustered hash table server will be deleted from the cluster.

The clustered hash table server was started using the Start Clustered Hash Table Server (STRCHTSVR) command. If the clustered hash table server has any active connections, any future requests from those connections fail.

### Restrictions:

- Cluster Resource Services must be active on the local node.
- All nodes specified in the NODE parameter must have Cluster Resource Services active.
- If an authorization list was specified when the server was started, the requesting user must have change (\*CHANGE) authority to the authorization list to end the server.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SERVER	Server	<i>Communications name</i>	Required, Positional 1
NODE	Node	Single values: *ALL Other values (up to 20 repetitions): <i>Communications name</i> , <u>*LOCAL</u>	Optional

Top

---

## Server (SERVER)

Specifies the clustered hash table server to be ended.

This is a required parameter.

**name** Specify the name of the clustered hash table server to be ended.

Top

---

## Node (NODE)

Specifies which nodes will end the clustered hash table server. The nodes specified will be removed from the clustered hash table domain. If all nodes in the clustered hash table domain are specified, the server will no longer exist in the cluster. Nodes in this list must be unique. The nodes must be active in the cluster.

### \*LOCAL

The clustered hash table server will be ended on the local node only. \*LOCAL can be specified only once in the list of nodes specified.

**\*ALL** The clustered hash table server will be ended on all cluster nodes in the clustered hash table domain. If specified, \*ALL must be the only value in the list.

*name* Specify the name of the nodes to process the end request. Up to 20 cluster nodes can be specified.

Top

---

## Examples

### Example 1: Ending a Clustered Hash Table Server on the Local Node

```
ENDCHTSVR  SERVER(CT0)
```

This command ends the clustered hash table server CT0 on the local node only.

### Example 2: Ending a Clustered Hash Table Server on One of two Nodes in the Clustered Hash Table Domain

Domain for clustered hash table CT1 is FRED and BARNEY.

```
ENDCHTSVR  SERVER(CT1)  NODE(FRED)
```

This command ends the clustered hash table server CT1 on cluster node FRED. The clustered hash table server is still active on BARNEY.

### Example 3: Ending a Clustered Hash Table Server on All Nodes

Domain for clustered hash table CT2 is FRED and BARNEY.

```
ENDCHTSVR  SERVER(CT2)  NODE(*ALL)
```

This command ends the clustered hash table server named CT2 on the local node (i.e. BARNEY) and node FRED. The clustered hash table will not exist in the cluster after this command runs.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFBD03

End clustered hash table server failed.

Top



---

## End Cleanup (ENDCLNUP)

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

Parameters  
Examples  
Error messages

The End Cleanup (ENDCLNUP) command allows you to end the cleanup operation. The cleanup operation allows items on the system to be deleted automatically after they are a specified number of days old. Any active batch cleanup jobs, either processing or on the job queue, are ended immediately.

This command does not alter any of the parameters specified on the Change Cleanup (CHGCLNUP) command. The cleanup operation can be restarted by specifying the Start Cleanup (STRCLNUP) command.

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

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

There are no parameters for this command.

Top

---

### Parameters

None

Top

---

### Examples

ENDCLNUP

This command ends the cleanup operation.

Top

---

### Error messages

#### \*ESCAPE Messages

##### CPF1E2A

Unexpected error in QSYSSCD job.

##### CPF1E2B

Power scheduler and cleanup options not found.

##### CPF1E33

Cleanup options or power schedule in use by another user.

##### CPF1E35

Not authorized to end cleanup.

**CPF1E36**

Cleanup has not been started.

**CPF1E99**

Unexpected error occurred.

[Top](#)

---

## End Cluster Node (ENDCLUNOD)

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

Parameters  
Examples  
Error messages

The End Cluster Node (ENDCLUNOD) command is used to end Cluster Resource Services on one or all the nodes in the membership list of an existing cluster. The status of each node that is ended is set to Inactive. In order to restart Cluster Resource Services on nodes that have been ended, the Start Cluster Node (STRCLUNOD) command is used.

When a node in the cluster is ended, it is not removed from the cluster membership list.

This command can be called on the node which is to be ended, or it can be called on any node in the cluster which has a status of Active. If this command is called when the cluster is partitioned, only nodes in the partition running the command will process the request.

The cluster resource group exit program on the node being ended will be called with an action code of End Node. The exit program on all other nodes in the recovery domain will be called with an action code of Failover. If all the nodes in the cluster are being ended, cluster resource group exit programs will not be called with an indication to failover.

The recovery domain of cluster resource groups on the node that had ended will indicate a node status of Active even though the node is inactive. For all the other nodes in the recovery domain, the status of the node will be Inactive. If the node being ended is the primary node for an active device cluster resource group, ownership of the hardware associated with the cluster resource group will be moved to a backup node. If the cluster resource group is not active, there are no backup nodes, or all backup nodes are either inactive or in a different cluster partition, the ownership of the hardware is left with the node being ended.

### Restrictions:

1. You must have input/output system configuration (\*IOSYSCFG) special authority to run this command.
2. This command cannot be called from a cluster resource group exit program.
3. The node being ended must be active.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CLUSTER	Cluster	<i>Name</i>	Required, Positional 1
NODE	Node identifier	<i>Name</i> , *ALL	Required, Positional 2
OPTION	Option	<u>*IMMED</u> , *CNTRLD	Optional

Top

---

## Cluster (CLUSTER)

Specifies the cluster that contains the node or nodes to be ended.

This is a required parameter.

*name* Specify the name of the cluster.

Top

---

## Node identifier (NODE)

Specifies the node identifier(s) to be ended.

This is a required parameter.

**\*ALL** End all active nodes in the cluster.

*name* Specify the name of the node to be ended.

Top

---

## Option (OPTION)

Specifies the method to end the node.

### **\*IMMED**

Immediate. The request to end Cluster Resource Services on the node will be processed immediately.

### **\*CNTRLD**

Controlled. Pending cluster resource group actions will complete before the request to end Cluster Resource Services is processed.

Top

---

## Examples

```
ENDCLUNOD CLUSTER(MYCLUSTER) NODE(NODE01) OPTION(*IMMED)
```

This command ends Cluster Resource Services on node NODE01 for cluster MYCLUSTER. The request is processed immediately without waiting for any pending cluster resource group actions to complete.

Top

---

## Error messages

Unknown

Top

---

## End Communications Server (ENDCMNSVR)

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

Parameters  
Examples  
Error messages

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

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

Top

---

### Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	*CNTRLD, *IMMED	Optional
DELAY	Controlled end delay time	1-86400, *NOMAX	Optional

Top

---

### How to end (OPTION)

Specifies whether the target display station pass-through server is ended in an immediate or controlled manner.

The possible values are:

#### \*CNTRLD

The server is ended in a controlled manner. Active sessions are allowed to complete their processing. New sessions are not allowed. After the specified period of time elapses, the processing for ENDCMNSVR OPTION(\*IMMED) is performed.

#### \*IMMED

The server is ended in an immediate fashion. All active sessions that were started through the target display station pass-through server are ended immediately.

Top

---

### Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) allowed in which to complete a controlled end of the target display station pass-through server. After this period of time all the target display station pass-through server jobs are ended immediately.

The possible values are:

#### \*NOMAX

There is no maximum amount of time to wait. The servers will not end until all active sessions end normally.

*delay-time*

Specify the number of seconds in which the end operation is completed. Valid values range from 1 through 86400 seconds.

Top

---

## Examples

### Example 1: Ending Target Display Station Pass-through Server

ENDCMNSVR

This command ends the target display station pass-through server in a controlled manner. Any active sessions that are using the target display station pass-through server are not affected. New sessions are not allowed through the target display station pass-through server. Once all of the active sessions have ended, the target display station pass-through server will end.

Top

---

## Error messages

None

Top

---

## End Communications Trace (ENDCMNTRC)

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

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

The End Communications Trace (ENDCMNTRC) command ends the trace running on the specified line, network interface, or network server description.

### Restrictions:

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

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
CFGOBJ	Configuration object	<i>Name</i>	Required, Positional 1
CFGTYPE	Type	*LIN, *NWI, *NWS	Required, Positional 2

[Top](#)

---

## Configuration object (CFGOBJ)

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

This is a required parameter.

*name* Specify the name of the configuration object for which communications tracing is to be ended.

[Top](#)

---

## Type (CFGTYPE)

Specifies the type of configuration description being traced.

This is a required parameter.

**\*LIN** The type of configuration object is a line description.

**\*NWI** The type of configuration object is a network interface description.

\*NWS The type of configuration object is a network server description.

Top

---

## Examples

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

This command ends the communications trace of line description QESLINE.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2601

Line description &1 not found.

#### CPF2634

Not authorized to object &1.

#### CPF26AE

Network server description &1 not found.

#### CPF39AE

Trace already ended.

#### CPF39AF

Trace is ending - please wait

#### CPF39A7

Trace storage not available in communications processor

#### CPF39A8

Not authorized to communications trace service tool

#### CPF39A9

Error occurred during communications trace function

#### CPF39BD

Network interface description &1 not found

#### CPF39B0

No communications traces exist.

#### CPF39B1

Trace &1 type &2 does not exist

#### CPF39B6

Communications trace function cannot be performed

#### CPF39C3

Trace &1 type &2 cannot be ended.

#### CPF98A2

Not authorized to &1 command.

Top



---

# End Commitment Control (ENDCMTCTL)

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

Parameters  
Examples  
Error messages

The End Commitment Control (ENDCMTCTL) command ends the commitment definition associated with the activation group for the program that issued the command. Changes to commitment resources associated with the commitment definition are no longer made after this command is processed.

This command either ends the activation group level or the job level commitment definition associated with the activation group for the program that issued the command. A commitment definition is first established by the Start Commitment Control (STRCMTCTL) command.

If there are uncommitted changes for an interactive job, a message is sent asking the user whether the changes should be committed or rolled back before a commitment definition is ended. For a batch job, the changes are rolled back.

More information on commitment control is in the "Commitment control" article in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

There are no parameters for this command.

Top

---

## Parameters

None

Top

---

## Examples

ENDCMTCTL

This command specifies that the commitment definition established with the STRCMTCTL command is to end. The system determines if any changes have been made to the commitment resources after the last commitment boundary (at the last completed Commit (COMMIT) command or Rollback (ROLLBACK) command). If changes have been made for an interactive job, a message is sent asking the user whether the changes should be made permanent (committed) or removed (rolled back). For batch jobs, any changes are rolled back.

Top

---

## Error messages

### \*ESCAPE Messages

**CPF83E4**

Commitment control ended with resources not committed.

**CPF835A**

End of commitment definition &1 canceled.

**CPF835B**

Errors occurred while ending commitment control.

**CPF835C**

Commitment control ended with remote changes not committed.

**CPF8350**

Commitment definition not found.

**CPF8355**

ENDCMTCTL not allowed. Pending changes active.

**CPF8356**

Commitment control ended with &1 local changes not committed.

**CPF8367**

Cannot perform commitment control operation.

[Top](#)

---

## End Copy Screen (ENDCPYSCN)

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

Parameters  
Examples  
Error messages

The End Copy Screen (ENDCPYSCN) command ends the copy screen image operation for the specified display device.

**Note:** The target display station can also stop the copy screen image operation if the user presses the System Request key and types ENDCPYSCN on the command line. No parameters can be specified.

Top

---

### Parameters

Keyword	Description	Choices	Notes
SRCDEV	Source device	Name, <u>*REQUESTER</u>	Optional, Positional 1

Top

---

### Source device (SRCDEV)

Specifies the display device that is currently having its screen images copied.

#### \*REQUESTER

Ends the copy screen image operation for the display device running this command.

*name* Specifies the device name of the display station that is having its screen images copied.

Top

---

### Examples

```
ENDCPYSCN SRCDEV(CHARLIE)
```

The command sends a message to 'CHARLIE' (the source display station). The message indicates the copy screen image operation is about to end. The target work station display is restored to the same display image that was shown before the operation started. The sign-on display is normally shown.

Top

---

### Error messages

#### \*ESCAPE Messages

##### CPF2207

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

##### CPF7AF7

Device name &1 not correct.

**CPF7AF8**

Device name &1 not being copied.

[Top](#)

---

## End Cluster Resource Group (ENDCRG)

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

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

The End Cluster Resource Group (ENDCRG) command disables resiliency of the specified cluster resource group. The cluster resource group status is set to Inactive if the resiliency is ended on all nodes in the recovery domain. The resources associated with the cluster resource group are no longer resilient. That is, there is no failover or switchover action provided for these resources while the cluster resource group is ended.

Ending a device cluster resource group will not change the ownership of devices. The devices remain on whatever nodes owns them at the time the command is run. Also, the devices are not varied off when the cluster resource group is ended.

Ending a peer cluster resource group will end the access point for the cluster resources on all nodes defined as a peer node.

If an exit program is specified for the cluster resource group, it is called with an action code of End on each active node in the recovery domain. When the exit program is called, the cluster resource group status is set to End Pending. Successful completion of the exit program sets the cluster resource group status to Inactive. In addition, for an application cluster resource group:

1. The current exit program job on the primary node will be cancelled with the \*IMMED option.
2. The takeover IP interface for the cluster resource group will be ended for the application cluster resource group.

If the exit program fails and the original state of the cluster resource group cannot be recovered, the cluster resource group status is set to Indoubt.

[Top](#)

---

### Parameters

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

[Top](#)

---

### Cluster (CLUSTER)

Specifies the cluster containing the cluster resource group.

This is a required parameter.

*name* Specify the name of the cluster.

---

## Cluster resource group (CRG)

Specifies the cluster resource group to be ended.

This is a required parameter.

*name* Specify the name of the cluster resource group to end.

Top

---

## Exit program data (EXITPGMDTA)

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

### \*SAME

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

### *character-value*

Specify the data that is to be passed to the exit program.

Top

---

## Examples

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

This command ends resiliency of the cluster resource group called MYCRG in the cluster called MYCLUSTER. When the cluster resource group exit program is called, it will be passed the exit program data 'important information' on all active nodes in the recovery domain.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1999

Errors occurred on command.

Top

---

# End Controller Recovery (ENDCTRLCY)

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

Parameters  
Examples  
Error messages

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

Use the Resume Controller Recovery (RSMCTRLCY) command to reestablish error recovery procedures for the controller.

[Top](#)

---

## Parameters

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

[Top](#)

---

## Controller (CTL)

Specifies the controller whose recovery is to be ended.

This is a required parameter.

[Top](#)

---

## Examples

```
ENDCTRLCY CTL(TROLL3)
```

This command ends error recovery procedures for the controller TROLL3.

[Top](#)

---

## 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.

**CPF5935**

Error occurred during command processing.

**CPF5936**

Not authorized to controller &1.

[Top](#)



---

## End Debug Mode (ENDDBG)

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

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

The End Debug (ENDDBG) command ends debug mode for a job, removes all breakpoints and traces, clears any trace data, and removes all programs from debug mode. This command cannot be entered when one or more of the programs in the call stack are stopped at a breakpoint. All breakpoints must be canceled by Resume Breakpoint (RSMBKP) or End Request (ENDRQS) commands. After this command has been entered, all database files in production libraries can be updated normally.

If ENDDBG is not done before the job has ended, all trace data is printed.

**Restriction:** This command is valid only in debug mode. To start debug mode, refer to the STRDBG (Start Debug) command.

If you are servicing another job and you are operating in debug mode, this command must be specified before you can use the End Service Job (ENDSRVJOB) command.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

ENDDBG

Assuming that this command is entered interactively and no program in the call stack is stopped at a breakpoint, debug mode for the job is ended.

[Top](#)

---

### Error messages

#### \*ESCAPE Messages

##### CPF1931

Command not valid at this time.

##### CPF1999

Errors occurred on command.

[Top](#)



---

## End Debug Server (ENDDBGSVR)

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

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

The End Debug Server (ENDDBGSVR) command ends the debug server router function. If there are active server jobs running when the router function is ended, the servers remain active until the connection with the client is ended. Subsequent connection requests fail until the debug server router function is started again.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

ENDDBGSVR

This command ends the debug server router function.

[Top](#)

---

### Error messages

None

[Top](#)



---

## End Database Monitor (ENDDBMON)

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

Parameters  
Examples  
Error messages

The End Database Monitor (ENDDBMON) command ends the collection of database performance statistics for a specified job, all jobs on the system or a selected set of jobs (i.e. a generic job name).

To end a monitor, you can specify the job or the monitor ID or both. If just the JOB parameter is specified, we end the monitor that was started using the same exact JOB parameter - if there is only one monitor which matches the specified JOB. If more than one monitor is active which matches the specified JOB, then the user uniquely identifies which monitor is to be ended by use of the MONID parameter. When just the MONID parameter is specified, we compare the specified MONID to the monitor ID of the monitor for the current job and to the monitor ID of all active public monitors (monitors that are open across multiple jobs). The monitor matching the specified MONID is closed.

The monitor ID is returned in the informational message CPI436A which is generated for each occurrence of the STRDBMON command. The monitor ID can also be found in column QQC101 of the QQQ3018 database monitor record.

### Restrictions:

- You cannot end database monitoring for a specific job by using JOB(\*ALL) on the ENDDBMON command. To end a specific job you must specify that job on the JOB parameter or specify JOB(\*).
- If JOB(\*ALL) was specified on the Start Database Monitor (STRDBMON) command, you cannot end monitoring on a per-job basis. Specifying ENDDBMON JOB(\*ALL) will end the monitor that is active across all jobs. Only if a specific job was specified on the JOB parameter or JOB(\*) can you end monitoring for a specific job.
- This command is conditionally threadsafe. It is not threadsafe (and may fail) when the OUTFILE parameter for the STRDBMON command specified a distributed file or a Distributed Data Management (DDM) file of type \*SNA.

Top

---

## Parameters

Keyword	Description	Choices	Notes
JOB	Job name	Single values: * Other values: <i>Qualified job name</i>	Optional, Positional 1
	Qualifier 1: Job name	<i>Generic name, name, *ALL</i>	
	Qualifier 2: User	<i>Generic name, name, *ALL</i>	
	Qualifier 3: Number	000000-999999, *ALL	
MONID	Monitor ID	<i>Character value, *ALL</i>	Optional
COMMENT	Comment	<i>Character value, *BLANK</i>	Optional

Top

---

## Job name (JOB)

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

### Single values

**\*** The data monitor for the job running the ENDDDBMON command is to be ended.

**\*ALL** The data monitor open across all jobs (started with JOB(\*ALL)) is to be ended.

### Qualifier 1: Job name

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

#### *generic-name*

Specify the generic job name of the monitor to be ended. The monitor started with the same generic job name (e.g. JOB(QZDA\*)) will be ended.

### Qualifier 2: User

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

#### *generic-name*

Specify the generic job user of the monitor to be ended. The monitor started with the same generic job user (e.g. JOB(\*ALL/DEVLPR\*)) will be ended.

### Qualifier 3: Number

**000000-999999**

Specify the number of the job whose database monitor is to be ended.

Top

---

## Monitor ID (MONID)

Specifies the unique monitor ID of the database monitor to be ended. If multiple monitors using the same JOB parameter were started, the MONID parameter must be specified to uniquely identify which individual monitor is to be ended. A monitor ID value is generated by the system for each invocation of the STRDBMON command. Informational message CPI436A contains the system generated monitor ID value. The system generated monitor ID value is also stored in column QQC101 of the QQQ3018 database monitor record.

**\*ALL** When used with JOB(\*ALL), all public monitors (monitors over multiple jobs) will be ended. When used with a generic job name (e.g. JOB(QZDA\*)), all public monitors (monitors over multiple jobs) started with JOB(QZDA\*) will be ended.

#### *simple-name*

Specify the 10-byte monitor ID of the monitor that is to be ended.

Top

---

## Comment (COMMENT)

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

## **\*BLANK**

Text is not specified.

## ***character-value***

Specify up to 100 characters of text.

Top

---

## **Examples**

### **Example 1: End Monitoring for a Specific Job**

```
ENDDBMON JOB(*)
```

This command ends database monitoring for the current job.

### **Example 2: End Monitoring for All Jobs**

```
ENDDBMON JOB(*ALL)
```

This command ends the monitor open across all jobs on the system. If more than one monitor with JOB(\*ALL) is active, then the MONID parameter must also be specified to uniquely identify which individual monitor to end.

### **Example 3: End Monitoring for an Individual Public Monitor with MONID Parameter**

```
ENDDBMON JOB(*ALL) MONID(061601001)
```

This command ends the monitor that was started with JOB(\*ALL) and that has a monitor ID of 061601001. Because there were multiple monitors started with JOB(\*ALL), the monitor ID must be specified to uniquely identify which monitor started with JOB(\*ALL) is to be ended.

### **Example 4: End Monitoring for an Individual Public Monitor with MONID Parameter**

```
ENDDBMON MONID(061601001)
```

The command performs the same function as the previous example. It ends the monitor that was started with JOB(\*ALL) or JOB(\*) and that has a monitor ID of 061601001.

### **Example 5: End Monitoring for All JOB(\*ALL) Monitors**

```
ENDDBMON JOB(*ALL/*ALL/*ALL) MONID(*ALL)
```

This command ends all monitors that are active across multiple jobs. It will not end any monitors open for a specific job or the current job.

### **Example 6: End Monitoring for a Generic Job**

```
ENDDBMON JOB(QZDA*)
```

This command ends the monitor that was started with JOB(QZDA\*). If more than one monitor with JOB(QZDA\*) is active, then the MONID parameter must also be specified to uniquely identify which individual monitor to end.

### **Example 7: End Monitoring for an Individual Monitor with a Generic Job**

```
ENDDBMON JOB(QZDA*) MONID(061601001)
```

This command ends the monitor that was started with JOB(QZDA\*) and has a monitor ID of 061601001. Because there were multiple monitors started with JOB(QZDA\*), the monitor ID must be specified to uniquely identify which JOB(QZDA\*) monitor is to be ended.

### Example 8: End Monitoring for a Group of Generic Jobs

```
ENDDBMON JOB(QZDA*) MONID(*ALL)
```

This command ends all monitors that were started with JOB(QZDA\*).

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF1321

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

#### CPF436D

Job &1 is not being monitored.

#### CPF436E

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

#### CPF43A1

Job &1 specified on ENDDBMON command is not unique.

[Top](#)



---

# End Device Recovery (ENDDEVRCY)

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

Parameters  
Examples  
Error messages

The End Device Recovery (ENDDEVRCY) command ends automatic error recovery procedures for a specific device. If any type of failure occurs after this command is run, an inquiry message is sent to the system operator. The user must have object operational authority for the device.

Use the Resume Device Recovery (RSMDEVRCY) command to reestablish error recovery procedures for the device.

[Top](#)

---

## Parameters

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

[Top](#)

---

## Device (DEV)

Specifies the device whose recovery is to be ended. Specify the name specified for the device in the device description.

[Top](#)

---

## Examples

```
ENDDEVRCY DEV(WSPR03)
```

This command ends error recovery procedures for the device WSPR03.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF5923

Device &1 does not allow automatic error recovery.

#### CPF5925

Device &1 not varied on.

#### CPF5935

Error occurred during command processing.

**CPF9814**

Device &1 not found.

[Top](#)

---

## End Directory Shadowing (ENDDIRSHD)

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

Parameters  
Examples  
Error messages

The End Directory Shadowing (ENDDIRSHD) command ends the directory shadow controlling job in the system work subsystem (QSYSWRK).

Any active collector or supplier jobs running are allowed to complete. No new collector jobs are started. Supplier jobs are prevented from starting if a collector system requests data through directory shadowing. The Start Directory Shadowing (STRDIRSHD) command can be used to re-start directory shadowing.

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

Top

---

### Parameters

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

Top

---

### How to end (OPTION)

Specifies whether the directory shadow controlling job is ended in a controlled manner or immediately.

#### \*CNTRLD

The directory shadow controlling job is ended in a controlled manner. This allows the directory shadow controlling job to perform cleanup (end-of-job processing).

#### \*IMMED

The directory shadow controlling job is ended immediately. The directory shadow controlling job is not allowed to perform any cleanup.

**Note:** Using the \*IMMED option can cause unexpected results if data has been only partially updated.

This is a required parameter.

Top

---

### Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) allowed for the directory shadow controlling job to complete its cleanup processing during a controlled end. This parameter is not valid if OPTION(\*IMMED) is specified. If the cleanup is not complete before the end of the delay time, the directory shadow controlling job is immediately ended.

**30** A maximum delay time of 30 seconds is allowed for cleanup before the directory shadow controlling job is ended.

*delay-time*

Specify the maximum amount of delay time in seconds before the directory shadow controlling job is ended. Valid values range from 1 through 999999.

This is a required parameter.

Top

---

## Examples

### Example 1: Ending Directory Shadowing in a Controlled Manner

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

The directory shadow controlling job is ended in the system work subsystem in a controlled manner and will have 60 seconds to complete its end-of-job processing.

### Example 2: Ending Directory Shadowing Immediately

```
ENDDIRSHD OPTION(*IMMED)
```

The directory shadow controlling job is ended in the system work subsystem immediately. The directory shadow controlling job does not perform end-of-job processing.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF89A9

Unable to end job that controls directory shadowing.

Top

---

## End Do Group (ENDDO)

### Where allowed to run:

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

**Threadsafe:** Yes

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

The End Do (ENDDO) command is used with the DO command to identify a group of commands that are processed together as a group. The ENDDO command specifies the end of the Do group that is started with an associated DO command. The ENDDO command must be specified after the last command in the Do group.

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

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

There are no parameters for this command.

[Top](#)

---

## Parameters

None

[Top](#)

---

## Examples

### Example 1: Processing a Group of Commands Unconditionally

```
DO
: (group of CL commands)
ENDDO
```

The commands between the DO and ENDDO commands are processed once, as a group of commands.

### Example 2: Processing a Group of Commands Conditionally

```
IF &SWITCH DO
: (group of CL commands)
ENDDO
```

The commands between the DO and ENDDO commands are processed if the value in the logical variable &SWITCH is '1'. If &SWITCH is not '1', then control passes immediately to the next command following the ENDDO command.

[Top](#)

---

## Error messages

None



---

## End Disk Reorganization (ENDDSKRGZ)

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

Parameters  
Examples  
Error messages

The End Disk Reorganization (ENDDSKRGZ) command allows the user to end the disk reorganization function started using the Start Disk Reorganization (STRDSKRGZ) CL command. The user can select to end disk reorganization for all auxiliary storage pools (ASPs) or for one or more specific ASPs. A message is sent to the system history (QHST) log when the reorganization function is ended for each ASP.

**Restriction:** You must have \*ALLOBJ special authority to use this command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
ASP	ASP number	Single values: *ALL Other values (up to 32 repetitions): 1-32	Optional, Positional 1
ASPDEV	ASP device	Values (up to 32 repetitions): <i>Name</i> , *ALLAVL	Optional

Top

---

### Auxiliary storage pool ID (ASP)

Specifies for which auxiliary storage pools the disk reorganization function is to be ended. A value must be specified for the ASP parameter or the ASPDEV parameter.

**\*ALL** Disk reorganization will be ended for the system ASP (ASP number 1) and all basic ASPs (ASP numbers 2-32) defined to the system.

*auxiliary-storage-pool-number*

Specify the ASP for which disk reorganization is to be ended. Valid ASP numbers are 1 to 32. Up to 32 ASP numbers may be specified.

Top

---

### ASP device (ASPDEV)

Specifies the name of the auxiliary storage pool (ASP) device for which the disk reorganization is to be ended. A value must be specified for the ASP parameter or the ASPDEV parameter.

**\*ALLAVL**

Disk reorganization will be ended for all ASP devices that currently have a status of 'Available'.

*auxiliary-storage-device-name*

Specify the name of the independent ASP device for which disk reorganization is to be ended. Up to 32 ASP device names may be specified.

Top

---

## Examples

### Example 1: Ending Disk Reorganization for ASP 1

```
ENDDSKRGZ  ASP(1)
```

This command allows the user to end the disk reorganization function for ASP 1.

### Example 2: Ending Disk Reorganization for All ASPs

```
ENDDSKRGZ  ASP(*ALL)
```

This command allows the user to end the reorganization function for each ASP that is currently being reorganized.

### Example 3: Ending Disk Reorganization for All ASP Devices

```
ENDDSKRGZ  ASPDEV(*ALLAVL)
```

This command allows the user to end the reorganization function for each ASP device that is currently being reorganized.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF1889

Disk reorganization not active for ASP &1

#### CPF1890

\*ALLOBJ authority required for requested operation.

[Top](#)



---

## End EPM Environments (ENDEPMENV)

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

Parameters  
Examples  
Error messages

The End EPM Environment (ENDEPMENV) command is used with extended program model (EPM) languages to delete a user-controlled environment. The C/400\*, FORTRAN/400\*, and Pascal languages are part of the extended program model.

You can use this command to delete a run-time environment for an EPM language application that you created with the STREPMENV command. See the Extended Program Model User's Guide and Reference for more detailed information on the EPM and this command.

---

### Error messages for ENDEPMENV

None

Top

---

### Parameters

Keyword	Description	Choices	Notes
EPMENV	Environment Name	<i>Character value</i> , *INACT, *MAIN, *REENT	Required, Positional 1
ENVNBR	Environment Number	1-65535	Optional, Positional 2

Top

---

### Environment Name (EPMENV)

Specifies which user-controlled environment is to be deleted. You must specify one of the special values (\*INACT, \*REENT, or \*MAIN) or an environment name. There is no default for this parameter.

#### **\*INACT**

All EPM user-controlled environments that are not invoked are deleted.

#### **\*REENT**

The reentrant environment with the corresponding ENVNBR is deleted. You must specify a value for ENVNBR if you specify this parameter.

#### **\*MAIN**

The \*MAIN user-controlled environment is deleted.

#### *environment-name*

Enter the name of the environment that you want to delete. The environment name must not begin with an asterisk (\*).

Top

---

## Environment Number (ENVNBR)

Specifies the environment number for the \*REENT environment that is to be deleted. This parameter is only valid if \*REENT is specified on EPMENV.

### *environment-number*

Enter the number of the reentrant environment.

[Top](#)

---

## Examples

None

[Top](#)

---

## Error messages

None

[Top](#)

---

## End Group Job (ENDGRPJOB)

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

Parameters  
Examples  
Error messages

The End Group Job (ENDGRPJOB) command ends a single job within a group and resumes another job within the group. You can specify the following:

- Which job in the group is ended
- Which job in the group gains control (this is valid only when a job is ending itself)
- Whether a job log is created for the job being ended

Top

---

### Parameters

Keyword	Description	Choices	Notes
GRPJOB	Group job	Name, *	Optional, Positional 1
RSMGRPJOB	Group job to be resumed	Name, *PRV	Optional
LOG	Job log	*NOLIST, *LIST	Optional

Top

---

### Group job (GRPJOB)

Specifies the group job name of the job being ended.

**\*** The group job that issued this command is ended.  
**\_**  
**name** Specify the group job name of the job being ended.

Top

---

### Group job to be resumed (RSMGRPJOB)

Specifies the group job name of the job that is resumed after the active job in the group has ended. This parameter is valid only when the job that issues this command is ending itself.

**\*PRV** The most recently active group job is resumed.  
**name** Specify the group job name of the job that is resumed after the active job in the group ends.

Top

---

### Job log (LOG)

Specifies whether to produce the job log for the ending group job.

**\*NOLIST**  
The information in the job log is not spooled to an output queue.

**\*LIST** The information in the job log is spooled to an output queue.

Top

---

## Examples

### Example 1: Ending Group Job that Issued Command

```
ENDGRPJOB GRPJOB(*) LOG(*LIST) RSMGRPJOB(GROUPJOB1)
```

This command ends the job that is currently running. Its job log is spooled to an output file for printing. When the job completes running, group job GROUPJOB1 becomes the active job in the group.

### Example 2: Printing Output of Ended Job

```
ENDGRPJOB GRPJOB(GROUPJOB2) LOG(*LIST)
```

Assume that the job issuing the ENDGRPJOB command is group job GROUPJOB1, which wants to end GROUPJOB2. Group job GROUPJOB2 ends. Its job log is spooled to an output file for printing.

### Example 3: Ending a Job That's Part of a Secondary Job Pair

```
ENDGRPJOB GRPJOB(*) LOG(*NOLIST)
```

Assume that the job issuing the ENDGRPJOB command is the only job in the group and is part of a secondary job pair. The job issuing the command ends. The job's job log is not spooled to an output file. When the job ends, the other job in the secondary job pair is resumed.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1309

Subsystem cannot complete the &1 command.

#### CPF1314

Value &1 for parameter &2 not allowed.

#### CPF1317

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

#### CPF1322

The End Group Job command not allowed at this time.

#### CPF1323

Group job &1 not ended; parameters do not agree.

#### CPF1324

Group job &1 not ended; parameters do not agree.

#### CPF1325

Group job &1 not ended; group job &2 does not exist.

#### CPF1326

Group job &1 does not exist.

#### CPF1327

Cannot end group job &1 with ENDGRPJOB.

**CPF1351**

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

Top



---

## End Host Server (ENDHOSTSVR)

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

Parameters  
Examples  
Error messages

The End Host Server (ENDHOSTSVR) command is used to end the optimized host server daemons. One or more server daemons can be ended and the server mapper daemon can be ended. Optionally, active connections to the \*DATABASE and \*FILE servers can be ended with this command.

If a server daemon is ended, and there are servers of that type that have active connections to client applications, the server jobs will remain active until communication with the client application is ended, unless the optional ENDACTCNN parameter is specified. Subsequent connection requests from the client application to that server daemon will fail however until the server daemon is started again.

If the server mapper daemon is ended, any existing client connections to the server jobs are unaffected. Subsequent requests from a client application to connect to the server mapper daemon (to obtain a server's port number) will fail however until the server mapper is started again.

A request to end \*ALL host server daemons will end any active daemons.

The ENDACTCNN parameter may be specified in order to end active connections to the \*DATABASE and \*FILE servers. This will cause the server jobs which are servicing these connections to be ended. The active connections can only be ended if the corresponding daemon job is also being ended. If the \*DATABASE keyword is specified, the QZDASOINIT and QZDASSINIT jobs which have active connections will be ended. If the \*FILE keyword is specified, the QPWFSEVSOS and QPWFSEVSOS jobs which have active connections will be ended.

---

## Error messages for ENDHOSTSVR

None

Top

---

## Parameters

Keyword	Description	Choices	Notes
SERVER	Server type	Single values: *ALL Other values (up to 8 repetitions): *CENTRAL, *DATABASE, *DTAQ, *FILE, *NETPRT, *RMTCMD, *SIGNON, *SVRMAP	Required, Positional 1
ENDACTCNN	End active connections	Single values: *NONE Other values (up to 2 repetitions): *DATABASE, *FILE	Optional, Positional 2

Top

---

## Server type (SERVER)

Specifies the server daemons to be ended.

The possible values are:

**\*ALL** All of the server daemons and the server mapper daemon are ended.

**\*CENTRAL**

The central server daemon in the QSYSWRK subsystem, if active, is ended.

**\*DATABASE**

The database server daemon in the QSERVER subsystem, if active, is ended.

**\*DTAQ**

The data queue server daemon in the QSYSWRK subsystem, if active, is ended.

**\*FILE** The file server daemon in the QSERVER subsystem, if active, is ended.

**\*NETPRT**

The network print server daemon in the QSYSWRK subsystem, if active, is ended.

**\*RMTCMD**

The remote command and distributed program call server daemon in the QSYSWRK subsystem, if active, is ended.

**\*SIGNON**

The signon server daemon in the QSYSWRK subsystem, if active, is ended.

**\*SVRMAP**

The server mapper daemon in the QSYSWRK subsystem, if active, is ended.

Top

---

## End active connections (ENDACTCNN)

Specifies whether or not the active connections for the specified servers will be ended.

### Single Value

**\*NONE:**

No active connections will be ended.

### Specific Server Values

**\*DATABASE:**

The active connections being serviced by the QZDASOINIT and QZDASSINIT server jobs will be ended. The server jobs servicing these connections will be ended.

**\*FILE:** The active connections being serviced by the QPWFSERVSO and QPWFSERVSS server jobs will be ended. The server jobs servicing these connections will be ended.

Top

---

## Examples

None

Top

---

## Error messages

None

Top



---

## End Input (ENDINP)

### Where allowed to run:

- Batch job (\*BATCH)

**Threadsafe:** No

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

The End Input (ENDINP) command is a delimiter in a batch input stream that indicates the end of the input data. The End Input (ENDINP) command also can indicate the end of an inline data file provided the command is detected while the inline file is being processed. If the inline file is using ending characters which are not defaults (//) the End Input (ENDINP) command is embedded without being recognized.

**Restrictions:** This command cannot be entered at a work station. Two slashes (//) in positions 1 and 2 must go before the End Input (ENDINP) command in the data record, for example //ENDINP. Blanks can separate the slashes from the command line (// ENDINP).

There are no parameters for this command.

[Top](#)

---

## Parameters

None

[Top](#)

---

## Examples

```
//BCHJOB  
:  
//DATA  
:  
//ENDINP
```

The ENDINP command indicates the end of a input stream that began with the Batch Job (BCHJOB) command.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

CPF1753

Command cannot be run.

[Top](#)



---

## End IP over SNA Interface (ENDIPSIFC)

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

Parameters  
Examples  
Error messages

The End IP over SNA Interface (ENDIPSIFC) command is used to end an AF\_INET sockets over SNA interface (an IP address by which this local host is known on the SNA transport).

**Note:** Ending an interface causes all routes associated with this interface to be deactivated immediately unless there are other active interfaces that the routes can switch to.

Top

---

### Parameters

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

Top

---

### Internet address (INTNETADR)

Specifies the internet address of an active (started) interface that had previously been added to the IP SNA configuration with the Add IP over SNA Interface (ADDIPSIFC) CL command. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

This is a required parameter.

Top

---

### Examples

```
ENDIPSIFC INTNETADR('9.5.1.248')
```

This command deactivates (ends) the interface with IP address 9.5.1.248.

Top

---

### Error messages

#### \*ESCAPE Messages

##### CPFA114

IP over SNA interface &1 not ended.

Top



---

## Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
500 Columbus Avenue  
Thornwood, NY8809  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) IBM 2006. Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. 1998, 2006. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing  
AFP  
AS/400  
CICS  
COBOL/400  
C/400  
DataPropagator  
DB2

IBM  
Infoprint  
InfoWindow  
iSeries  
LPDA  
OfficeVision  
i5/OS  
Print Services Facility  
RPG/400  
SystemView  
System/36  
TCS  
WebSphere

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

---

## Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

**Personal Use:** You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

**Commercial Use:** You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE









Printed in USA