



IBM Systems - iSeries

i5/OS Commands

Starting with RMVDEVDMNE (Remove Device Domain Entry)

*Version 5 Release 4*







IBM Systems - iSeries

i5/OS Commands

Starting with RMVDEVDMNE (Remove Device Domain  
Entry)

*Version 5 Release 4*

**Note**

Before using this information and the product it supports, be sure to read the information in "Notices," on page 385.

**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>Remove Device Domain Entry (RMVDEVDMNE)</b> . . . . .	<b>1</b>
Parameters . . . . .	1
Cluster (CLUSTER) . . . . .	1
Device domain (DEVDMN) . . . . .	2
Node identifier (NODE) . . . . .	2
Examples . . . . .	2
Error messages . . . . .	2
<b>Remove Directory (RMVDIR)</b> . . . . .	<b>3</b>
Parameters . . . . .	4
Directory (DIR) . . . . .	4
Remove link (RMVLNK) . . . . .	4
Examples . . . . .	4
Error messages . . . . .	4
<b>Remove Directory Entry (RMVDIRE)</b> . . . . .	<b>7</b>
Parameters . . . . .	7
User identifier (USRID) . . . . .	7
User description (USRD) . . . . .	8
Command character identifier (CMDCHRID) . . . . .	8
Examples . . . . .	9
Error messages . . . . .	9
<b>Remove Directory Shadow System (RMVDIRSHD)</b> . . . . .	<b>11</b>
Parameters . . . . .	11
System name (SYSNAME) . . . . .	11
Remove data (RMVDTA) . . . . .	11
Examples . . . . .	12
Error messages . . . . .	12
<b>Remove DLO Authority (RMVDLOAUT)</b> <b>13</b>	<b>13</b>
Parameters . . . . .	13
Document library object (DLO) . . . . .	13
Folder (FLR) . . . . .	14
User profile (USER) . . . . .	14
Authorization list (AUTL) . . . . .	14
Document access code (ACC) . . . . .	14
System object name (SYSOBJNAM) . . . . .	14
Examples . . . . .	15
Error messages . . . . .	15
<b>Remove Distribution List Entry (RMVDSTLE)</b> . . . . .	<b>17</b>
Parameters . . . . .	17
List identifier (LSTID) . . . . .	17
User identifier (USRID) . . . . .	18
Command character identifier (CMDCHRID) . . . . .	18
Examples . . . . .	19
Error messages . . . . .	19

<b>Remove Distribution Queue (RMVDSTQ)</b> . . . . .	<b>21</b>
Parameters . . . . .	21
Distribution queue (DSTQ) . . . . .	21
Examples . . . . .	22
Error messages . . . . .	22
<b>Remove Distribution Route (RMVDSTRTE)</b> . . . . .	<b>23</b>
Parameters . . . . .	23
System name (SYSNAME) . . . . .	23
Examples . . . . .	24
Error messages . . . . .	24
<b>Remove Secondary System Name (RMVDSTSYSN)</b> . . . . .	<b>25</b>
Parameters . . . . .	25
Secondary system name (SYSNAME) . . . . .	25
Examples . . . . .	25
Error messages . . . . .	26
<b>Remove Configuration Entry (RMVEMLCFGE)</b> . . . . .	<b>27</b>
Parameters . . . . .	27
Configuration entry (EMLCFGE) . . . . .	27
Examples . . . . .	27
Error messages . . . . .	27
<b>Remove Environment Variable (RMVENVVAR)</b> . . . . .	<b>29</b>
Parameters . . . . .	29
Environment variable (ENVVAR) . . . . .	29
Level (LEVEL) . . . . .	29
Examples . . . . .	30
Error messages . . . . .	30
<b>Remove EWC Barcode Entry (RMVEWCBCDE)</b> . . . . .	<b>31</b>
Parameters . . . . .	31
Barcode group (BCDGRP) . . . . .	31
Initialization source member (INZMBR) . . . . .	31
Initialization source file (INZFILE) . . . . .	31
Examples . . . . .	32
Error messages . . . . .	32
<b>Remove EWC PTC Entry (RMVEWCPTCE)</b> . . . . .	<b>33</b>
Parameters . . . . .	33
PTC group (PTCGRP) . . . . .	33
Initialization source member (INZMBR) . . . . .	33
Initialization source file (INZFILE) . . . . .	33
Examples . . . . .	34
Error messages . . . . .	34

**Remove Exit Program (RMVEXITPGM) 35**

Parameters . . . . . 35  
Exit point (EXITPNT) . . . . . 35  
Exit point format (FORMAT) . . . . . 35  
Program number (PGMNBR) . . . . . 35  
Examples . . . . . 36  
Error messages . . . . . 36

**Remove Font Table Entry (RMVFNTTBLE) . . . . . 37**

Parameters . . . . . 37  
Font table (FNTTBL) . . . . . 38  
Printer to host font (PHFCS) . . . . . 39  
Printer to host code page (PHCP) . . . . . 40  
Host to printer font (HPFCS) . . . . . 41  
Host to printer code page (HPCP) . . . . . 41  
Printer to printer font (PPFCS) . . . . . 41  
Examples . . . . . 42  
Error messages . . . . . 42

**Remove Filter Action Entry (RMVFTRACNE) . . . . . 45**

Parameters . . . . . 45  
Filter (FILTER) . . . . . 45  
Group (GROUP) . . . . . 45  
Examples . . . . . 45  
Error messages . . . . . 46

**Remove Filter Selection Entry (RMVFTRSLTE) . . . . . 47**

Parameters . . . . . 47  
Filter (FILTER) . . . . . 47  
Sequence number (SEQNBR) . . . . . 47  
Examples . . . . . 48  
Error messages . . . . . 48

**Remove ICF Device Entry (RMVICFDEVE) . . . . . 49**

Parameters . . . . . 49  
ICF communication file (FILE) . . . . . 49  
Program device (PGMDEV) . . . . . 49  
Examples . . . . . 50  
Error messages . . . . . 50

**Remove Image Catalog Entry (RMVIMGCLGE) . . . . . 51**

Parameters . . . . . 51  
Image catalog (IMGCLG) . . . . . 51  
Image catalog index (IMGCLGIDX) . . . . . 51  
Image catalog volume (VOL) . . . . . 52  
Keep image files (KEEP) . . . . . 52  
Examples . . . . . 52  
Error messages . . . . . 52

**Remove IP over SNA Interface (RMVIPSIFC) . . . . . 55**

Parameters . . . . . 55  
Internet address (INTNETADR) . . . . . 55

Examples . . . . . 55  
Error messages . . . . . 56

**Remove IP over SNA Location (RMVIPSLOC) . . . . . 57**

Parameters . . . . . 57  
Remote destination (RMTDEST) . . . . . 57  
Subnet mask (SUBNETMASK) . . . . . 57  
Examples . . . . . 58  
Error messages . . . . . 58

**Remove IP over SNA Route (RMVIPS RTE) . . . . . 59**

Parameters . . . . . 59  
Route destination (RTEDEST) . . . . . 59  
Subnet mask (SUBNETMASK) . . . . . 59  
Next hop (NEXTHOP) . . . . . 60  
Examples . . . . . 60  
Error messages . . . . . 60

**Remove Job Queue Entry (RMVJOBQE) 61**

Parameters . . . . . 61  
Subsystem description (SBSD) . . . . . 61  
Job queue entry job queue (JOBQ) . . . . . 62  
Examples . . . . . 62  
Error messages . . . . . 62

**Remove Job Schedule Entry (RMVJOBSCDE) . . . . . 63**

Parameters . . . . . 63  
Job name (JOB) . . . . . 63  
Entry number (ENTRYNBR) . . . . . 64  
Examples . . . . . 64  
Error messages . . . . . 64

**Remove Journalized Changes (RMVJRNCHG) . . . . . 67**

Parameters . . . . . 68  
Journal (JRN) . . . . . 69  
Journalized file identification (FILE) . . . . . 70  
Objects (OBJ) . . . . . 71  
Range of journal receivers (RCVRNG) . . . . . 72  
Starting large sequence number (FROMENTLRG) . . . . . 73  
Ending large sequence number (TOENTLRG) . . . . . 73  
Fully qualified job name (TOJOBQ) . . . . . 74  
Commitment boundary (CMTBDY) . . . . . 75  
Option (OPTION) . . . . . 75  
Object error option (OBJERROPT) . . . . . 75  
Output (OUTPUT) . . . . . 76  
File to receive output (OUTFILE) . . . . . 76  
Output member options (OUTMBR) . . . . . 76  
Type of output information (DETAIL) . . . . . 77  
Starting sequence number (FROMENT) . . . . . 77  
Ending sequence number (TOENT) . . . . . 78  
Examples . . . . . 78  
Error messages . . . . . 79

**Remove LAN Adapter Information (RMVLANADPI) . . . . . 83**

Parameters . . . . .	83
Adapter (ADPTNAME) . . . . .	83
Adapter address (ADPTADR) . . . . .	83
Examples . . . . .	83
Error messages . . . . .	84

**Remove LAN Adapter (RMVLANADPT) 85**

Parameters . . . . .	85
Line description (LINE) . . . . .	85
Adapter (ADPTNAME) . . . . .	85
Adapter address (ADPTADR) . . . . .	86
Examples . . . . .	86
Error messages . . . . .	86

**Remove Library List Entry (RMVLIBLE) 87**

Parameters . . . . .	87
Library (LIB) . . . . .	87
Examples . . . . .	87
Error messages . . . . .	87

**Remove License Key Information (RMVLICKEY) . . . . . 89**

Parameters . . . . .	89
License key input (LICKEYINP) . . . . .	89
Product identifier (PRDID) . . . . .	90
License term (LICTRM) . . . . .	90
Feature (FEATURE) . . . . .	90
System serial number (SERIAL) . . . . .	90
License key file (LICKEYFILE) . . . . .	91
License key member (LICKEYMBR) . . . . .	91
Examples . . . . .	91
Error messages . . . . .	92

**Remove Link (RMVLNK) . . . . . 93**

Parameters . . . . .	94
Object link (OBJLNK) . . . . .	94
Examples . . . . .	94
Error messages . . . . .	94

**Remove Member (RMVM) . . . . . 97**

Parameters . . . . .	97
Data base file (FILE) . . . . .	97
Member (MBR) . . . . .	98
Examples . . . . .	98
Error messages . . . . .	98

**Remove Mounted FS (RMVMFS) . . . . . 101**

Parameters . . . . .	101
Type of file system (TYPE) . . . . .	101
Directory mounted over (MNTOVRDIR) . . . . .	102
Mounted file system (MFS) . . . . .	102
Examples . . . . .	102
Error messages . . . . .	102

**Remove Message (RMVMSG) . . . . . 103**

Parameters . . . . .	103
Call stack entry message queue (PGMQ) . . . . .	103
Message queue (MSGQ) . . . . .	105
Message key (MSGKEY) . . . . .	106

Clear (CLEAR) . . . . .	106
Remove unhandled exception (RMVEXCP) . . . . .	106
Reject default reply (RJTDFTRPY) . . . . .	107
Examples . . . . .	107
Error messages . . . . .	107

**Remove Message Description (RMVMSGD) . . . . . 109**

Parameters . . . . .	109
Message identifier (MSGID) . . . . .	109
Message file (MSGF) . . . . .	109
Examples . . . . .	110
Error messages . . . . .	110

**Remove Nickname (RMVNCK) . . . . . 111**

Parameters . . . . .	111
Nickname (NCK) . . . . .	111
Examples . . . . .	112
Error messages . . . . .	112

**Remove Network Job Entry (RMVNETJOBE) . . . . . 113**

Parameters . . . . .	113
User ID (FROMUSRID) . . . . .	113
Examples . . . . .	114
Error messages . . . . .	114

**Remove Network Table Entry (RMVNETTBLE) . . . . . 115**

Parameters . . . . .	115
Network (NETWORK) . . . . .	115
Internet address (INTNETADR) . . . . .	115
Examples . . . . .	115
Error messages . . . . .	116

**Remove Node List Entry (RMVNODLE) 117**

Parameters . . . . .	117
Node list (NODL) . . . . .	117
Remote location (RMTLOCNAME) . . . . .	118
SNA node name (CPNAME) . . . . .	118
Examples . . . . .	119
Error messages . . . . .	119

**Remove NetWare Aut Entry (RMVNTWAUTE) . . . . . 121**

Parameters . . . . .	121
Server type (SVRTYPE) . . . . .	121
NDS tree (NDSTREE) . . . . .	121
Server (SERVER) . . . . .	122
User profile (USRPRF) . . . . .	122
Examples . . . . .	122
Error messages . . . . .	122

**Remove Server Storage Link (RMVNWSSTGL) . . . . . 123**

Parameters . . . . .	123
Network server storage space (NWSSTG) . . . . .	123
Network server description (NWSD) . . . . .	123

Renumber link (RENUMBER) . . . . .	123
Examples . . . . .	124
Error messages . . . . .	124

**Remove Optical Cartridge (RMVOPTCTG) . . . . . 125**

Parameters . . . . .	125
Volume identifier (VOL) . . . . .	125
Volume description option (VOLOPT) . . . . .	126
Media location (MEDLOC) . . . . .	126
Removed cartridge location (RMVCTGLOC) . . . . .	126
Optical media library (MLB) . . . . .	127
Optical device (DEV) . . . . .	127
Examples . . . . .	127
Error messages . . . . .	127

**Remove Optical Server (RMVOPTSVR) 131**

Parameters . . . . .	131
Side information (CSI) . . . . .	131
Volume description option (VOLOPT) . . . . .	131
Examples . . . . .	132
Error messages . . . . .	132

**Remove Protocol Table Entry (RMVPCLTBLE) . . . . . 133**

Parameters . . . . .	133
Protocol (PROTOCOL) . . . . .	133
Examples . . . . .	133
Error messages . . . . .	133

**Remove PEX Definition (RMVPEXDFN) 135**

Parameters . . . . .	135
Definition (DFN) . . . . .	135
Examples . . . . .	136
Error messages . . . . .	136

**Remove PEX Filter (RMVPEXFTR) . . . 137**

Parameters . . . . .	137
Filter (FTR) . . . . .	137
Examples . . . . .	138
Error messages . . . . .	138

**Remove PF Constraint (RMVPCFCST) 139**

Parameters . . . . .	139
File (FILE) . . . . .	139
Constraint name (CST) . . . . .	140
Constraint type (TYPE) . . . . .	140
Remove constraint (RMVCST) . . . . .	141
Examples . . . . .	141
Error messages . . . . .	142

**Remove Physical File Trigger (RMVPFTRG) . . . . . 143**

Parameters . . . . .	143
Physical file (FILE) . . . . .	144
Trigger time (TRGTIME) . . . . .	144
Trigger event (TRGEVENT) . . . . .	144
Trigger (TRG) . . . . .	144
Trigger library (TRGLIB) . . . . .	145

Examples . . . . .	145
Error messages . . . . .	146

**Remove Program (RMVPGM) . . . . . 147**

Parameters . . . . .	147
Program (PGM) . . . . .	147
Examples . . . . .	147
Error messages . . . . .	148

**Remove Prestart Job Entry (RMVPJE) 149**

Parameters . . . . .	149
Subsystem description (SBSD) . . . . .	149
Program (PGM) . . . . .	150
Examples . . . . .	150
Error messages . . . . .	150

**Remove Program Temporary Fix (RMVPTF) . . . . . 151**

Parameters . . . . .	151
Product (LICPGM) . . . . .	151
Release (RLS) . . . . .	151
PTF numbers to select (SELECT) . . . . .	152
PTF numbers to omit (OMIT) . . . . .	152
Extent of change (RMV) . . . . .	152
Delayed PTFs (DELAYED) . . . . .	153
Remove on unattended IPL (IPLRMV) . . . . .	153
Remove dependent PTFs (RMVDEP) . . . . .	153
Examples . . . . .	154
Error messages . . . . .	154

**Remove RDB Directory Entry (RMVRDBDIRE) . . . . . 157**

Parameters . . . . .	157
Entry (RDB) . . . . .	157
Examples . . . . .	157
Error messages . . . . .	158

**Remove REXX Buffer (RMVREXBUF) 159**

Parameters . . . . .	159
Buffer (BUFFER) . . . . .	159
Examples . . . . .	159
Error messages . . . . .	160

**Remove Remote Definition (RMVRMTDFN) . . . . . 161**

Parameters . . . . .	161
System name (SYSTEM) . . . . .	161
Examples . . . . .	161
Error messages . . . . .	162

**Remove Remote Journal (RMVRMTJRN) . . . . . 163**

Parameters . . . . .	163
Relational database (RDB) . . . . .	164
Source journal (SRCJRN) . . . . .	164
Target journal (TGTJRN) . . . . .	164
Examples . . . . .	165
Error messages . . . . .	165



**Remove Reply List Entry (RMVRPYLE) 167**

Parameters . . . . .	167
Sequence number (SEQNBR) . . . . .	167
Examples . . . . .	168
Error messages . . . . .	168

**Remove Routing Entry (RMVRTGE) 169**

Parameters . . . . .	169
Subsystem description (SBSD) . . . . .	169
Routing entry sequence number (SEQNBR) . . . . .	170
Examples . . . . .	170
Error messages . . . . .	170

**Remove Search Index Entry (RMVSHIDX) 171**

Parameters . . . . .	171
Search index (SCHIDX) . . . . .	171
Panel group (PNLGRP) . . . . .	172
Examples . . . . .	172
Error messages . . . . .	172

**Remove Sphere of Control Entry (RMVSOCE) 173**

Parameters . . . . .	173
Entry (ENTRY) . . . . .	173
Examples . . . . .	173
Error messages . . . . .	174

**Remove Service Table Entry (RMVSRVTBLE) 175**

Parameters . . . . .	175
Service (SERVICE) . . . . .	175
Port (PORT) . . . . .	175
Protocol (PROTOCOL) . . . . .	175
Examples . . . . .	175
Error messages . . . . .	176

**Remove Server Auth Entry (RMVSVRAUTE) 177**

Parameters . . . . .	177
User profile (USRPRF) . . . . .	177
Server (SERVER) . . . . .	177
Examples . . . . .	178
Error messages . . . . .	178

**Remove Tape Cartridge (RMVTAPCTG) 179**

Parameters . . . . .	179
Library device (DEV) . . . . .	179
Cartridge ID (CTG) . . . . .	179
Category (CGY) . . . . .	180
Station (STATION) . . . . .	181
Examples . . . . .	181
Error messages . . . . .	182

**Remove TCP/IP Host Table Entry (RMVTCPHTE) 183**

Parameters . . . . .	183
----------------------	-----

Internet address (INTNETADR) . . . . .	183
Examples . . . . .	183
Error messages . . . . .	184

**Remove TCP/IP Interface (RMVTCPIFC) 185**

Parameters . . . . .	185
Internet address (INTNETADR) . . . . .	185
Alias name (ALIASNAME) . . . . .	186
Examples . . . . .	186
Error messages . . . . .	186

**Remove TCP/IP Port Restriction (RMVTCPPORT) 187**

Parameters . . . . .	187
Range of port values (PORT) . . . . .	187
Protocol (PROTOCOL) . . . . .	188
User profile (USRPRF) . . . . .	188
Examples . . . . .	188
Error messages . . . . .	189

**Remove Point-to-Point Profile (RMVTCPPTP) 191**

Parameters . . . . .	191
Configuration profile (CFGPRF) . . . . .	191
Delete communications config (DLTCMNCFG) . . . . .	191
Examples . . . . .	192
Error messages . . . . .	192

**Remove TCP/IP Remote System (RMVTCPRSI) 193**

Parameters . . . . .	193
Internet address (INTNETADR) . . . . .	193
Examples . . . . .	193
Error messages . . . . .	193

**Remove TCP/IP Route (RMVTCPRTE) 195**

Parameters . . . . .	195
Route destination (RTEDEST) . . . . .	195
Subnet mask (SUBNETMASK) . . . . .	196
Type of service (TOS) . . . . .	197
Next hop (NEXTHOP) . . . . .	197
Preferred binding interface (BINDIFC) . . . . .	197
Examples . . . . .	198
Error messages . . . . .	198

**Remove TCP/IP Server (RMVTCPSVR) 199**

Parameters . . . . .	199
Server special value (SVRSPCVL) . . . . .	199
Examples . . . . .	199
Error messages . . . . .	199

**Remove TCP/IP Table (RMVTCPTBL) 201**

Parameters . . . . .	201
Table - Help . . . . .	201
Examples . . . . .	201
Error messages . . . . .	201

**Remove Trace (RMVTRC) . . . . . 203**

Parameters . . . . .	203
Trace statements to remove (STMT) . . . . .	203
Program (PGM) . . . . .	204
Examples . . . . .	204
Error messages . . . . .	204

**Remove Trace Filter (RMVTRCFTR) 205**

Parameters . . . . .	205
Filter (FTR) . . . . .	205
Examples . . . . .	205
Error messages . . . . .	206

**Remove Work Station Entry (RMVWSE) . . . . . 207**

Parameters . . . . .	207
Subsystem description (SBSD) . . . . .	207
Work station name (WRKSTN) . . . . .	208
Work station type (WRKSTNTYPE) . . . . .	208
Examples . . . . .	209
Error messages . . . . .	209

**Rename Object (RNM). . . . . 211**

Parameters . . . . .	211
Object (OBJ) . . . . .	212
New object (NEWOBJ) . . . . .	212
Examples . . . . .	212
Error messages . . . . .	212

**Rename Directory Entry (RNMDIRE) 215**

Parameters . . . . .	215
Old user identifier (OLDUSRID) . . . . .	215
New user identifier (NEWUSRID) . . . . .	216
Forward from user identifier (FWDFRM) . . . . .	217
Change network user identifier (NETUSRID) . . . . .	217
Examples . . . . .	217
Error messages . . . . .	218

**Rename Document Library Object (RNMDLO). . . . . 219**

Parameters . . . . .	219
Document library object (DLO) . . . . .	219
New document library object (NEWDLO) . . . . .	219
In folder (FLR) . . . . .	219
Examples . . . . .	220
Error messages . . . . .	220

**Rename Distribution List (RNMDSTL) 221**

Parameters . . . . .	221
List identifier (LSTID) . . . . .	221
New list identifier (NEWLSTID) . . . . .	222
Examples . . . . .	222
Error messages . . . . .	222

**Rename LAN Adapter (RNMLANADPI) 225**

Parameters . . . . .	225
Adapter (ADPTNAME) . . . . .	225
New name (NEWNAME) . . . . .	225
Examples . . . . .	225

Error messages . . . . .	226
--------------------------	-----

**Rename Member (RNMM) . . . . . 227**

Parameters . . . . .	227
Data base file (FILE) . . . . .	227
Member (MBR) . . . . .	228
New member (NEWMBR) . . . . .	228
Examples . . . . .	228
Error messages . . . . .	228

**Rename Nickname (RNMNCK) . . . . . 229**

Parameters . . . . .	229
Nickname (NCK) . . . . .	229
New nickname (NEWNCK) . . . . .	230
Examples . . . . .	230
Error messages . . . . .	230

**Rename Object (RNMOBJ). . . . . 231**

Parameters . . . . .	233
Object (OBJ) . . . . .	233
Object type (OBJTYPE) . . . . .	233
New object (NEWOBJ) . . . . .	234
ASP device (ASPDEV) . . . . .	234
System (SYSTEM) . . . . .	235
Examples . . . . .	235
Error messages . . . . .	235

**Rename TCP/IP Host Table Entry (RNMTCPHTE) . . . . . 239**

Parameters . . . . .	239
Internet address (INTNETADR) . . . . .	239
New internet address (NEWINTNETA) . . . . .	240
Examples . . . . .	240
Error messages . . . . .	240

**Rollback (ROLLBACK) . . . . . 241**

Parameters . . . . .	241
Examples . . . . .	241
Error messages . . . . .	241

**Start RPC Binder Daemon (RPCBIND) 243**

Parameters . . . . .	243
Retrieve RPC registration (RTVRPCREG) . . . . .	243
Examples . . . . .	243
Error messages . . . . .	244

**Convert RPC Source (RPCGEN) . . . . . 245**

Parameters . . . . .	245
From file (FROMFILE) . . . . .	245
Option (OPTION) . . . . .	245
Protocol (PROTOCOL) . . . . .	246
To file (TOFILE) . . . . .	246
Examples . . . . .	247
Error messages . . . . .	247

**Replace Document (RPLDOC) . . . . . 249**

Parameters . . . . .	249
Document (TODOC) . . . . .	250
In folder (TOFLR) . . . . .	250

Document file (DOCFILE) . . . . .	250
Document member (DOCMBR) . . . . .	250
Document part (DOCPART) . . . . .	251
User identifier (USRID) . . . . .	251
Document identifier (DOCID) . . . . .	251
Document type (DOCTYPE) . . . . .	252
Document character identifier (DOCCHRID) . . . . .	252
Command character identifier (CMDCHRID) . . . . .	253
Examples . . . . .	253
Error messages . . . . .	253

**Request Order Assistance (RQSORDAST) . . . . . 255**

Parameters . . . . .	255
Examples . . . . .	255
Error messages . . . . .	255

**Reroute Job (RRTJOB) . . . . . 257**

Parameters . . . . .	257
Routing data (RTGDTA) . . . . .	257
Request data or command (RQSDTA) . . . . .	258
Examples . . . . .	258
Error messages . . . . .	258

**Resume Breakpoint (RSMBKP) . . . . . 259**

Parameters . . . . .	259
Examples . . . . .	259
Error messages . . . . .	259

**Resume Controller Recovery (RSMCTLCY) . . . . . 261**

Parameters . . . . .	261
Controller (CTL) . . . . .	261
Examples . . . . .	262
Error messages . . . . .	262

**Resume Device Recovery (RSMDEVRCY) . . . . . 263**

Parameters . . . . .	263
Device (DEV) . . . . .	263
Examples . . . . .	264
Error messages . . . . .	264

**Resume Line Recovery (RSMLINRCY) 265**

Parameters . . . . .	265
Line (LINE) . . . . .	265
Examples . . . . .	265
Error messages . . . . .	265

**Resume NWI Recovery (RSMNWIRCY) 267**

Parameters . . . . .	267
Network interface description (NWI) . . . . .	267
Examples . . . . .	267
Error messages . . . . .	267

**Restore Object (RST) . . . . . 269**

Parameters . . . . .	269
Device (DEV) . . . . .	270

Objects (OBJ) . . . . .	270
Name pattern (PATTERN) . . . . .	271
Directory subtree (SUBTREE) . . . . .	272
Output (OUTPUT) . . . . .	272
Create parent directories (CRTPRNDIR) . . . . .	273
Parent directory owner (PRNDIROWN) . . . . .	273
Volume identifier (VOL) . . . . .	273
Label (LABEL) . . . . .	274
Sequence number (SEQNBR) . . . . .	274
End of media option (ENDOPT) . . . . .	274
Optical file (OPTFILE) . . . . .	275
Type of output information (INFTYPE) . . . . .	275
System (SYSTEM) . . . . .	275
Date when saved (SAVDATE) . . . . .	275
Time when saved (SAVTIME) . . . . .	276
Option (OPTION) . . . . .	276
Allow object differences (ALWOBJDIF) . . . . .	276
Force object conversion (FRCOBJCVN) . . . . .	278
Object ID (OBJID) . . . . .	278
Examples . . . . .	278
Error messages . . . . .	280

**Restore Authority (RSTAUT) . . . . . 283**

Parameters . . . . .	284
User profile (USRPRF) . . . . .	284
Saved from ASP device (SAVASPDEV) . . . . .	284
Restore to ASP device (RSTASPDEV) . . . . .	285
Examples . . . . .	285
Error messages . . . . .	286

**Restore Configuration (RSTCFG) . . . . . 287**

Parameters . . . . .	287
Objects (OBJ) . . . . .	288
Device (DEV) . . . . .	289
Object types (OBJTYPE) . . . . .	289
Volume identifier (VOL) . . . . .	290
Sequence number (SEQNBR) . . . . .	291
End of media option (ENDOPT) . . . . .	291
Save file (SAVF) . . . . .	291
System Resource Management (SRM) . . . . .	292
Optical file (OPTFILE) . . . . .	292
Allow object differences (ALWOBJDIF) . . . . .	292
Output (OUTPUT) . . . . .	293
File to receive output (OUTFILE) . . . . .	294
Output member options (OUTMBR) . . . . .	294
Objects to omit (OMITOBJ) . . . . .	295
Examples . . . . .	295
Error messages . . . . .	296

**Restore Document Lib Object (RSTDLO) . . . . . 299**

Parameters . . . . .	300
Document library object (DLO) . . . . .	301
Saved folder (SAVFLR) . . . . .	301
New object name (RENAME) . . . . .	302
Restore into folder (RSTFLR) . . . . .	302
Device (DEV) . . . . .	302
System object name (SYSOBJNAM) . . . . .	303
Volume identifier (VOL) . . . . .	303
Sequence number (SEQNBR) . . . . .	303

End of media option (ENDOPT)	304
Label (LABEL)	304
Save file (SAVF)	305
Optical file (OPTFILE)	305
Object name generation (NEWOBJ)	305
Date when saved (SAVDATE)	306
Time when saved (SAVTIME)	306
Allow object differences (ALWOBJDIF)	306
Saved from ASP number (SAVASP)	307
Restore to ASP number (RSTASP)	307
Output (OUTPUT)	307
File to receive output (OUTFILE)	308
Output member options (OUTMBR)	308
Examples	309
Error messages	310

### Restore Library (RSTLIB) . . . . . 313

Parameters	314
Saved library (SAVLIB)	315
Device (DEV)	317
Volume identifier (VOL)	317
Sequence number (SEQNBR)	318
Label (LABEL)	318
End of media option (ENDOPT)	319
Starting library (STRLIB)	319
Save file (SAVF)	320
Media definition (MEDDFN)	320
Optical file (OPTFILE)	320
Libraries to omit (OMITLIB)	321
Option (OPTION)	321
Data base member option (MBROPT)	322
Spooled file data (SPLFDTA)	322
Date when saved (SAVDATE)	322
Time when saved (SAVTIME)	322
Allow object differences (ALWOBJDIF)	323
Force object conversion (FRCOBJCVN)	324
Restore to library (RSTLIB)	325
Restore to ASP device (RSTASPDEV)	325
Restore to ASP number (RSTASP)	326
Output (OUTPUT)	326
File to receive output (OUTFILE)	326
Output member options (OUTMBR)	327
Type of output information (INFTYPE)	327
Objects to omit (OMITOBJ)	328
Examples	328
Error messages	331

### Restore Licensed Program (RSTLICPGM) . . . . . 335

Parameters	335
Product (LICPGM)	336
Device (DEV)	336
Optional part to be restored (OPTION)	337
Type of object to be restored (RSTOBJ)	337
Language for licensed program (LNG)	338
Output (OUTPUT)	338
Release (RLS)	338
Replace release (REPLACERLS)	339
Volume identifier (VOL)	339
Sequence number (SEQNBR)	339

End of media option (ENDOPT)	340
Save file (SAVF)	340
Library (LIB)	341
Language library (LNGLIB)	341
Folder (FLR)	341
Code home directory (CODHOMEDIR)	341
Language home directory (LNGHOMEDIR)	342
Force object conversion (FRCOBJCVN)	342
Examples	343
Error messages	344

### Restore Object (RSTOBJ) . . . . . 345

Parameters	346
Objects (OBJ)	348
Saved library (SAVLIB)	349
Device (DEV)	349
Object types (OBJTYPE)	350
Volume identifier (VOL)	350
Sequence number (SEQNBR)	351
Label (LABEL)	352
End of media option (ENDOPT)	352
Save file (SAVF)	352
Option (OPTION)	353
File member (FILEMBR)	353
Data base member option (MBROPT)	354
Spooled file data (SPLFDTA)	354
Date when saved (SAVDATE)	354
Time when saved (SAVTIME)	355
Allow object differences (ALWOBJDIF)	355
Force object conversion (FRCOBJCVN)	356
Restore to library (RSTLIB)	357
Output (OUTPUT)	357
Media definition (MEDDFN)	358
Optical file (OPTFILE)	358
Restore to ASP device (RSTASPDEV)	359
Restore to ASP number (RSTASP)	359
File to receive output (OUTFILE)	359
Output member options (OUTMBR)	360
Type of output information (INFTYPE)	360
Libraries to omit (OMITLIB)	360
Objects to omit (OMITOBJ)	361
Examples	362
Error messages	362

### Restore S/36 File (RSTS36F) . . . . . 367

Parameters	369
To file (TOFILE)	369
To library (TOLIB)	369
Device (DEV)	370
Set identifier (SET)	370
User specified DBCS data (IGCDTA)	370
File label (FROMLABEL)	371
Creation date (CRTDATE)	371
Sequence number (SEQNBR)	371
Volume identifier (VOL)	371
End of tape option (ENDOPT)	371
Physical file (PHYFILE)	372
Data base member option (MBROPT)	372
Date differentiated file (DATDIFF)	372
Examples	373

Error messages . . . . . 374

**Restore S/36 Library Members  
(RSTS36LIBM) . . . . . 377**

Parameters . . . . . 378  
To member (TOMBR). . . . . 379  
To library (TOLIB). . . . . 379  
Device (DEV) . . . . . 379  
S/36 source members (SRCMBRS) . . . . . 379  
Data base member option (MBROPT) . . . . . 380  
User specified DBCS data (IGCDTA) . . . . . 380  
File label (FROMLABEL) . . . . . 380

Creation date (CRTDATE) . . . . . 380  
Sequence number (SEQNBR) . . . . . 380  
Volume identifier (VOL). . . . . 381  
End of tape option (ENDOPT). . . . . 381  
Physical file (PHYFILE) . . . . . 381  
S/36 object members (OBJMBRS). . . . . 382  
Examples. . . . . 382  
Error messages . . . . . 382

**Appendix. Notices . . . . . 385**

Trademarks . . . . . 386  
Terms and conditions. . . . . 387



---

# Remove Device Domain Entry (RMVDEVDMNE)

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

Parameters  
Examples  
Error messages

The Remove Device Domain Entry (RMVDEVDMNE) command is used to remove a cluster node from a device domain. The removed node will no longer be able to be in the recovery domain for a device cluster resource group.

A node which has been removed from a device domain will most likely need to be IPLed before it can be added to any device domain. One example of this situation would be if a device description for an auxiliary storage pool has been varied on since the last IPL.

This command can be called on any node in the cluster which has a status of Active.

## 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 to be removed cannot be in the recovery domain of any device cluster resource group.
4. The node to be removed and at least one other member of the device domain must be active. On certain conditions, all current members of the device domain must be active.
5. This command will fail if any member of the device domain from which the node being removed has a status of Partition.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CLUSTER	Cluster	<i>Name</i>	Required, Positional 1
DEVDMN	Device domain	<i>Name</i>	Required, Positional 2
NODE	Node identifier	<i>Name</i>	Required, Positional 3

Top

---

## Cluster (CLUSTER)

Specifies the cluster that contains the node.

This is a required parameter.

*name* Specify the name of the cluster which contains the node.

Top

---

## Device domain (DEVDMN)

Specifies the device domain from which the node is to be removed.

This is a required parameter.

*name* Specify the name of the device domain from which the node is to be removed.

Top

---

## Node identifier (NODE)

Specifies the cluster node which will be removed from the device domain.

This is a required parameter.

*name* Specify the name of the node.

Top

---

## Examples

```
RMVDEVDMNE CLUSTER(MYCLUSTER) DEVDMN(MYDOMAIN) NODE(NODE01)
```

This command removes the node NODE01 from the device domain MYDOMAIN in cluster MYCLUSTER. If the cluster membership list is NODE01, NODE02, and NODE03 and all three nodes were in the same device domain, then nodes NODE02 and NODE03 still belong to device domain MYDOMAIN.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1999

Errors occurred on command.

Top



---

## Remove Directory (RMVDIR)

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

Parameters  
Examples  
Error messages

The Remove Directory (RMVDIR) command removes a specified directory from the system after all objects in the directory have been unlinked and the directory is no longer in use. If a directory to be removed contains objects, this command optionally unlinks all of the objects and then deletes the directory. If the user does not have the authority to unlink every object in the directory, only those objects for which the user has the authority are unlinked. When an object cannot be unlinked, the directory and all objects in the directory that cannot be unlinked are not removed.

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

- RD
- RMDIR

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

### Restrictions:

1. In the "root" (/), QOpenSys and user-defined file systems, the user must have object existence (\*OBJEXIST) authority for the specified directory, and \*OBJEXIST authority for every object in it. If the user does not have \*OBJEXIST authority for one or more objects in the directory, those objects are not unlinked and the directory is not removed.
2. In the "root" (/), QOpenSys, and user-defined file systems, the user must have write, execute (\*WX) authority to the parent directory.
3. In the QDLS file system, the user must have all (\*ALL) authority to the directory and execute (\*X) authority to its parent directory.
4. The user must have execute (\*X) authority to the prefix directory.
5. See the iSeries Security Reference, SC41-5302 book for the authority requirements for other file systems.
6. A user cannot remove a directory within a "root" (/), QOpenSys, or user-defined file system directory that has the "restricted rename and unlink" attribute set on (this attribute is equivalent to the S\_ISVTX mode bit) unless one or more of the following are true:
  - a. The user is the owner of the directory to be removed.
  - b. The user is the owner of the parent directory of the directory to be removed.
  - c. The user has all object (\*ALLOBJ) special authority.
7. A directory can not be removed if it is the current directory for a job.
8. This command cannot be used to delete reserved directories and libraries.
9. When an object is in use in QSYS.LIB, independent ASP QSYS.LIB, or QDLS, the object cannot be unlinked. When an object is in use in QOpenSys or the "root" (/) file system, the object is successfully unlinked, and the object is deleted when no longer in use.

Top

---

## Parameters

Keyword	Description	Choices	Notes
DIR	Directory	<i>Path name</i>	Required, Positional 1
RMVLNK	Remove link	<u>*NO</u> , *YES	Optional

Top

---

## Directory (DIR)

Specifies the path name of the directory or a pattern to match the path name or names of directories to be removed.

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

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Top

---

## Remove link (RMVLNK)

Specifies whether to unlink all objects in a directory or not allow the directory to be deleted if it contains objects.

- \*NO Only an empty directory is removed. A directory may contain entries for the directory (.) and for the parent directory (..) and still be treated as an empty directory.
- \*YES All objects within the specified directory are deleted. If the file system that contains the directory does not support removal of links in the directory, error message CPFA0AC "Request cannot be completed. Directory contains objects." will be sent.

Top

---

## Examples

### Example 1: Removing a Directory and the Objects in that Directory

```
RMVDIR DIR('W') RMVLNK(*YES)
```

This command removes directory W after all of its objects have been unlinked. If directory W contains objects and you have the authority to unlink all of those objects, all of the objects are unlinked and directory W is removed. If you do not have authority to unlink all of the objects, only those for which you have authority are unlinked and the directory is not removed.

Top

---

## Error messages

### \*ESCAPE Messages

**CPFA085**  
Home directory not found for user &1.

**CPFA093**  
Name matching pattern not found.

**CPFA09C**  
Not authorized to object. Object is &1.

**CPFA09D**  
Error occurred in program &1.

**CPFA09E**  
Object in use. Object is &1.

**CPFA0A1**  
An input or output error occurred.

**CPFA0A3**  
Path name resolution causes looping.

**CPFA0A7**  
Path name too long.

**CPFA0A9**  
Object not found. Object is &1.

**CPFA0AB**  
Operation failed for object. Object is &1.

**CPFA0AC**  
Request cannot be completed. Directory contains objects.

**CPFA0AD**  
Function not supported by file system.

**CPFA0B1**  
Requested operation not allowed. Access problem.

**CPFA0B2**  
No objects satisfy request.

**CPFA0B7**  
&1 directories removed. &2 directories not removed.

Top



---

## Remove Directory Entry (RMVDIRE)

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

Parameters  
Examples  
Error messages

The Remove Directory Entry (RMVDIRE) command allows you to remove a specific user entry from the system distribution directory. When a user ID and address is removed from the directory, it is also removed from any distribution lists. If a user ID and address has multiple descriptions associated with it, options exist to remove only a specific description or all descriptions.

If an \*ANY user is removed and an \*ANY \*ANY directory entry exists, the user is not removed from the distribution lists but the description is changed to the \*ANY \*ANY description.

### Restrictions:

1. You must have security administrator (\*SECADM) special authority.
2. The user entry being removed must not have ownership of documents or folders in the local Document Interchange Architecture (DIA) library.
3. If the user entry being removed has incoming distributions on its distribution queue and the remove request was submitted interactively, an inquiry message appears asking whether the mail should be removed. If the answer is no, the user entry is not removed from the directory. If the command was submitted in a batch program, the mail is not cleared and the user entry is not removed from the directory.

Top

---

## Parameters

Keyword	Description	Choices	Notes
USRID	User identifier	<i>Element list</i>	Required, Positional 1
	Element 1: User ID	<i>Character value</i>	
	Element 2: Address	<i>Character value</i>	
USRD	User description	<i>Character value, *FIRST, *ALL</i>	Optional
CMDCHRID	Command character identifier	Single values: <i>*SYSVAL, *DEVVD</i> Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	

Top

---

## User identifier (USRID)

Specifies the user ID and address for the user entry being removed. Both parts must be provided. A maximum of 8 characters can be entered for each field.

Top

---

## User description (USRD)

Specifies the description associated with the user ID and address. Since more than one entry can exist in the directory for a specific user ID and address, the description fully defines the user entry being removed.

### \*FIRST

The first entry in the directory for the specified user ID and address is removed. If only one entry exists, it is removed.

**\*ALL** All descriptions for the given user ID and address are removed and the user ID and address is removed.

### *user-description*

Specify a maximum of 50 characters for the description for the user. This must exactly match the description as it exists in the directory for this user ID and address to be removed, or an error message is returned.

Top

---

## Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values.

### Note:

- Only the user ID and address, system name and group, department, and the X.400 O/R parameters are translated to the graphic character set identifier (GCID) specified on this parameter. All other parameter values that you specify are stored exactly as they are entered; the GCID value is stored with them.
- If this command is run interactively, the default GCID value is taken from the display device description. If it is run in batch, the default GCID value is taken from the QCHRID system value. You can override these values by specifying a specific character set and code page on this parameter.

### Single values

#### \*SYSVAL

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

#### \*DEVDD

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

### Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

### Element 2: Code page

1-32767

Specify the code page to use.

Top

---

## Examples

RMVDIRE USRID(HURST NEWYORK) USRD('Manager of Payroll')

User ID and address HURST NEWYORK is removed if the following is true:

- An entry exists in the directory with the specified user ID, address, and description.
- The user does not own any documents or folders in the document interchange architecture (DIA) library.
- The user is not enrolled in the OfficeVision.
- The user has received all mail from the mail queue.

In addition, the user is removed from all distribution lists.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF89A3

Operation not successful due to authority reasons.

#### CPF90A8

Operation not successful due to data validation reasons.

#### CPF90A8

\*SECADM special authority required to do requested operation.

#### CPF9005

System resource required to complete this request not available.

#### CPF9009

System requires file &1 in &2 be journaled.

#### CPF9024

System cannot get correct record to finish operation.

#### CPF905C

Error occurred trying to find a translation table.

#### CPF9087

Directory entry not removed.

#### CPF9096

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

#### CPF9810

Library &1 not found.

#### CPF9838

User profile storage limit exceeded.

#### CPF9845

Error occurred while opening file &1.

#### CPF9846

Error while processing file &1 in library &2.

Top





---

## Remove Directory Shadow System (RMVDIRSHD)

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

Parameters  
Examples  
Error messages

The Remove Directory Shadow System (RMVDIRSHD) command removes a system that is currently shadowing directory data to the local system.

**Restriction:** To use this command, you must have security administrator (\*SECADM) authority.

Top

---

### Parameters

Keyword	Description	Choices	Notes
SYSNAME	System name	<i>Character value</i>	Required, Positional 1
RMVDTA	Remove data	*YES, *NO	Optional

Top

---

### System name (SYSNAME)

Specifies the name of the system for which shadowing is to be removed. The name can contain a maximum of eight alphanumeric characters. You can specify uppercase letters A through Z, numbers 0 through 9, and special characters @, #, \$, and embedded blanks. Embedded blanks must be enclosed in single quotation marks ('). Leading blanks are not allowed. The @, #, and \$ characters are not recommended because they are not part of an invariant character set and are not available on all keyboards.

This is a required parameter.

Top

---

### Remove data (RMVDTA)

This parameter specifies whether to remove directory data received from the system that is being removed.

**\*NO** Directory data that has been previously shadowed is left on the local system. Modifications are not made to this data through shadowing unless you shadow data from another system that has shadowed data from the system being removed.

**\*YES** Directory entry data that was shadowed from the system specified is removed from the local system. Department and location data is not removed.

Top

---

## Examples

```
RMVDIRSHD  SYSNAME(NYCITY)  RMVDTA(*YES)
```

This command removes the system NYCITY from shadowing and removes all the data shadowed from NYCITY.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF89A5

Remove or suspend of shadow supplier &1 was not successful.

#### CPF90A8

\*SECADM special authority required to do requested operation.

#### CPF905C

Error occurred trying to find a translation table.

#### CPF9838

User profile storage limit exceeded.

#### 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

---

## Remove DLO Authority (RMVDLOAUT)

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

Parameters  
Examples  
Error messages

The Remove Document Library Object Authority (RMVDLOAUT) command allows you to remove an existing user authority for documents or folders.

The following types of authority can be removed:

- An existing specific user's authority
- The authorization list's authority for an object
- An existing access code

### Restrictions:

You must have all (\*ALL) authority to the objects or all object (\*ALLOBJ) special authority, or be the owner of the objects to use this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Character value, *SYSOBJNAM, *ALL	Required, Positional 1
FLR	Folder	Character value, *NONE	Optional
USER	User profile	Single values: *SAME, *ALL Other values (up to 50 repetitions): <i>Element list</i>	Optional, Positional 2
	Element 1: User profile	<i>Name</i>	
AUTL	Authorization list	<i>Name</i> , *SAME	Optional
ACC	Document access code	Values (up to 50 repetitions): 0-2047, *SAME, *ALL	Optional
SYSOBJNAM	System object name	<i>Name</i>	Optional

Top

---

## Document library object (DLO)

Specifies the document or folder for which authority is removed.

**\*ALL** All objects in a specified folder have user authority removed. If \*ALL is specified, the **Folder (FLR)** parameter is required.

### \*SYSOBJNAM

The system object name specified on the **System object name (SYSOBJNAM)** parameter has user authority removed.

*name* Specify the user-assigned name of the document or folder for which user authority is removed. A maximum of 12 characters can be specified.

Top

---

## Folder (FLR)

Specifies the folder where the object specified is located. If \*SYSOBJNAM is specified on the **Document library object (DLO)** parameter, this parameter is ignored.

### \*NONE

A folder name is not specified. If a name is specified on the **Document library object (DLO)** parameter, and the object is located in a folder, then \*NONE cannot be specified. If \*ALL is specified on the **Document library object (DLO)** parameter, then \*NONE cannot be specified.

*name* Specify the user-assigned name of the folder where the object specified is located. 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

---

## User profile (USER)

Specifies the names of users whose specific authority is removed.

### \*SAME

The specific authority is not removed.

\*ALL All users' specific user authority is removed (except for the owner of the object).

*name* Specify the name of the user profile that has specific authority removed.

Top

---

## Authorization list (AUTL)

Specifies the name of the existing authorization list whose authority for the object is removed.

### \*SAME

The authority specified in the existing authorization list for the object is not removed.

*name* Specify the name of the existing authorization list whose authority for the object is removed.

Top

---

## Document access code (ACC)

Specifies the access codes to be removed for the object.

### \*SAME

No access codes for the object are removed.

\*ALL All access codes for the object are removed.

*0-2047* Specify the access codes which are to be removed. A maximum of 50 access codes can be specified.

Top

---

## System object name (SYSOBJNAM)

Specifies the system object name of the document or folder.

*name* Specify the system object name of the document or folder using the entire 10 characters.

Top

---

## Examples

```
RMVDLOAUT DLO(DOCA) FLR(MYFLR) AUTL(MYLIST)
```

This command removes the authority of the authorization list MYLIST for object DOCA in folder MYFLR.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF8A75

Not authorized to access folder &1.

#### 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.

#### CPF8A83

Not authorized to access document &2 in folder &1.

#### CPF8A88

Operation not allowed on document &2 in folder &1.

#### CPF8A89

Document &2 in folder &1 is logically damaged.

#### CPF90BA

Authority request for document library object failed.

#### CPF9073

No authority to view or change the security of document library object &1.

#### CPF908A

Requester &1 not enrolled.

#### CPF908B

Document library object not found.

#### CPF908E

&1 objects changed; &2 objects not changed.

#### CPF909A

Document &2 in folder &1 is damaged.

#### CPF9095

Folder &1 is damaged.



---

## Remove Distribution List Entry (RMVDSTLE)

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

Parameters  
Examples  
Error messages

The Remove Distribution List Entry (RMVDSTLE) command allows you to remove entries from a distribution list. Up to 300 entries can be removed from the list at one time.

**Restriction:** You must have security administrator authority (\*SECADM) to remove entries from a distribution list owned by someone else.

Top

---

### Parameters

Keyword	Description	Choices	Notes
LSTID	List identifier	<i>Element list</i>	Required, Positional 1
	Element 1: List ID	<i>Character value</i>	
	Element 2: List ID qualifier	<i>Character value</i>	
USRID	User identifier	Values (up to 300 repetitions): <i>Element list</i>	Required, Positional 2
	Element 1: User ID	<i>Character value</i>	
	Element 2: Address	<i>Character value</i>	
	Element 3: User description	<i>Character value, *FIRST, *ALL</i>	
CMDCHRID	Command character identifier	Single values: <i>*SYSVAL, *DEV</i> Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	

Top

---

### List identifier (LSTID)

Specifies the two-part list identification (ID) of the distribution list that is to have entries removed.

This is a required parameter.

#### *list-identification*

The list ID is entered in two parts separated by at least one space. If any lowercase characters are specified, the system changes them into uppercase letters.

If any lowercase characters are specified, the system changes them to, and stores them as, uppercase characters.

Top

---

## User identifier (USRID)

Specifies the user ID, address, and description of the user being removed from the distribution list. Both the user ID and address must be provided. The description can be entered to specify deletion of a specific description for a user ID.

You can enter multiple values for this parameter.

A two-part list ID (user-ID and address) can be used in place of the user ID and address to identify a remote distribution list being removed.

Up to 300 sets of user IDs, addresses, and descriptions can be specified. Each valid set is removed from the distribution list.

At least one set of user IDs is required.

### *user-id and address*

Specify the user ID and address being removed.

If any lowercase characters are specified, the system changes them to, and stores them as, uppercase characters.

### \*FIRST

The first entry in the distribution list for the specified user ID and address is removed. If only one entry exists, it is removed from the list.

**\*ALL** All the entries with the specified user ID and address are removed from the distribution list.

### *user-description*

Specify the description of the user. If a list ID is specified, enter the list description. The description can have a maximum of 50 characters.

Top

---

## Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values.

### Note:

- Only the user ID and address, system name and group, department, and the X.400 O/R parameters are translated to the graphic character set identifier (GCID) specified on this parameter. All other parameter values that you specify are stored exactly as they are entered; the GCID value is stored with them.
- If this command is run interactively, the default GCID value is taken from the display device description. If it is run in batch, the default GCID value is taken from the QCHRID system value. You can override these values by specifying a specific character set and code page on this parameter.

### Single values

#### \*SYSVAL

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

#### **\*DEV**

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



## Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

## Element 2: Code page

1-32767

Specify the code page to use.

### Note:

- Only the user ID and address, system name and group, department, and the X.400 O/R parameters are translated to the graphic character set identifier (GCID) specified on this parameter. All other parameter values that you specify are stored exactly as they are entered; the GCID value is stored with them.
- If this command is run interactively, the default GCID value is taken from the display device description. If it is run in batch, the default GCID value is taken from the QCHRID system value. You can override these values by specifying a specific character set and code page on this parameter.

Top

---

## Examples

```
RMVDSTLE  LSTID(CHICAGO DLIST)
           USRID((HURST PAYROLL 'Manager of Payroll')
                (LEE DEPT554 *FIRST)
                (BOCA DLIST 'Remote Distribution List')
                (BRYON WAREHSE *ALL))
```

In this example, four user IDs are removed from the distribution list CHICAGO DLIST. The third user ID is, in fact, a remote distribution list. All entries for BRYON WAREHSE are removed from the list.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9024

System cannot get correct record to finish operation.

#### CPF905C

Error occurred trying to find a translation table.

#### CPF9092

No entries removed from list &1 &2.

#### CPF9093

&1 entries removed from list &2 &3.

#### CPF9096

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

#### CPF9838

User profile storage limit exceeded.

#### CPF9845

Error occurred while opening file &1.

Top



---

## Remove Distribution Queue (RMVDSTQ)

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

Parameters  
Examples  
Error messages

The Remove Distribution Queue (RMVDSTQ) command allows you to remove a distribution queue entry from the distribution services queue table. Distribution queues are used to store distributions before they are sent or forwarded to other systems.

The RMVDSTQ command does not provide interactive display support. This is provided by the Configure Distribution Services (CFGDSTSRV) command. More information about configuring a distribution network is in the SNA Distribution Services book, SC41-5410.

Distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

### Restrictions:

- This command is shipped with public \*EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
- The following distribution queues cannot be removed:
  - Queues referred to in the routing table
  - Queues that contain distributions waiting to be sent
  - DLS (document library services) queues that have remote libraries configured to use them
  - SVDS (IBM System Manager for i5/OS distribution services) queues when a receiver is active or when distributions have been received and the sender has not acknowledged receiving confirmation.
- Messages that report errors about distribution queues may display or print different characters than you entered for the distribution queue name because of internal system transformations. Similarly (depending on the language used for the work station), the internal value for a distribution queue name may differ from the characters shown for the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the **Distribution queue** prompt (DSTQ parameter) does not match the rules for an internal distribution queue value or if it does not match the internal value for any defined distribution queue (ignoring case differences).

Top

---

## Parameters

Keyword	Description	Choices	Notes
DSTQ	Distribution queue	<i>Character value</i>	Required, Positional 1

Top

---

## Distribution queue (DSTQ)

Specifies the name of the distribution queue entry to be removed.

This is a required parameter.

---

## Examples

RMVDSTQ DSTQ(CHICAGO)

This command removes the distribution queue entry named CHICAGO.

---

## Error messages

### \*ESCAPE Messages

**CPF8802**

Distribution queue &1 was not found.

**CPF8807**

Error occurred while using QSNADS journal.

**CPF8809**

Errors detected on SNADS internal queues.

**CPF881B**

Distributions are being received for distribution queue &1

**CPF8826**

Distribution queue entries exist for distribution queue &1.

**CPF8827**

Routing table entries exist for distribution queue &1.

**CPF8828**

Remote document library entries exist for \*DLS distribution queue &1.

**CPF8833**

Distribution queue &1 already exists.

**CPF8849**

Queue &1 in use by another distribution services function.

**CPF9845**

Error occurred while opening file &1.

**CPF9846**

Error while processing file &1 in library &2.

**CPF9847**

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

**CPF9899**

Error occurred during processing of command.

---

## Remove Distribution Route (RMVDSTRTE)

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

Parameters  
Examples  
Error messages

The Remove Distribution Route (RMVDSTRTE) command allows you to remove an entry from the distribution services routing table. Once a system is removed from the table, distributions can no longer be sent directly to that system from this system.

Interactive display support is provided by the Configure Distribution Services (CFGDSTSRV) command. More information on configuring a distribution network is in the SNA Distribution Services book, SC41-5410.

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

System names and group names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Top

---

### Parameters

Keyword	Description	Choices	Notes
SYSNAME	System name	<i>Element list</i>	Required, Positional 1
	Element 1: System name	<i>Character value, *ANY</i>	
	Element 2: System group	<i>Character value, *ANY</i>	

Top

---

### System name (SYSNAME)

Specifies the system name and group name of the remote system you want to remove from the routing table.

You can specify a maximum of 8 characters for the system name and a maximum of 8 characters for the group name.

\*ANY can be specified for the system name. When SYSNAME(\*ANY group) is specified, you remove the routing table entry used to resolve a distribution destination that does not match a specific system name, but matches a group name. Only one \*ANY is allowed for each group in the table.

\*ANY can be specified for the group name only if \*ANY is also specified for the system name. When SYSNAME(\*ANY \*ANY) is specified, you remove the routing table entry used to resolve a distribution destination that does not match any other routing table entries. Only one SYSNAME(\*ANY \*ANY) entry is allowed in the table.

This is a required parameter.

---

## Examples

### Example 1: Removing a System from the Routing Table

```
RMVDSTRTE  SYSNAME(SYSTEMA GROUPA)
```

This command removes the routing table entry for the system named SYSTEMA.

### Example 2: Removing a Generic Routing Table Entry

```
RMVDSTRTE  SYSNAME(*ANY GROUPNM1)
```

This command removes a routing table entry that has a system name of \*ANY and a group name of GROUPNM1.

---

## Error messages

### \*ESCAPE Messages

#### CPF8802

Distribution queue &1 was not found.

#### CPF8807

Error occurred while using QSNADS journal.

#### CPF8815

Routing table entry &1 &2 not found.

#### CPF8831

Entry &1 &2 already exists in routing table.

#### CPF8837

System name/Group &1 &2 in use by another distribution services function.

#### CPF8849

Queue &1 in use by another distribution services function.

#### CPF9845

Error occurred while opening file &1.

#### CPF9846

Error while processing file &1 in library &2.

#### CPF9847

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

#### CPF9899

Error occurred during processing of command.

---

## Remove Secondary System Name (RMVDSTSYSN)

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

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

The Remove Distribution Secondary System Name (RMVDSTSYSN) command allows you to remove an entry from the distribution services secondary system name table. The table contains the names of all alternate (or alias) system names for which the local system receives and may redirect distributions. When an alternate system name is removed from the table, this system no longer receives distributions for the alternate system.

Interactive display support is provided by the Configure Distribution Services (CFGDSTSRV) command. More information about configuring a distribution network is in the SNA Distribution Services book, SC41-5410.

System names and group names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

### Restrictions:

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

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
SYSNAME	Secondary system name	<i>Element list</i>	Required, Positional 1
	Element 1: System name	<i>Character value</i>	
	Element 2: System group	<i>Character value</i>	

[Top](#)

---

## Secondary system name (SYSNAME)

Specifies the system name and group name of the secondary system name table entry to be removed.

You can specify a maximum of 8 characters for the system name and a maximum of 8 characters for the group name.

This is a required parameter.

[Top](#)

---

## Examples

```
RMVDSTSYSN  SYSNAME(SYS2LAJ1 ROCHESTR)
```

This command removes the system named SYS2LAJ1 ROCHESTR from the distribution services secondary system name table.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF8807

Error occurred while using QSNADS journal.

#### CPF8818

Secondary system name table entry &1 &2 not found.

#### CPF8835

System name/Group &1 &2 already specified.

#### CPF8837

System name/Group &1 &2 in use by another distribution services function.

#### 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.

#### CPF9899

Error occurred during processing of command.

Top



---

## Remove Configuration Entry (RMVEMLCFGE)

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

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

The Remove Emulation Configuration Entry (RMVEMLCFGE) command is used to remove a configuration entry for a 3270 device emulation session from the configuration file.

**Restriction:** You cannot remove the configuration entry QEMDFTCFGE, which is the default emulation configuration entry shipped with the system, with this command.

[Top](#)

---

### Parameters

Keyword	Description	Choices	Notes
EMLCFGE	Configuration entry	<i>Name</i>	Required, Positional 1

[Top](#)

---

### Configuration entry (EMLCFGE)

Specifies the name of the configuration entry you are removing.

This is a required parameter.

[Top](#)

---

### Examples

```
RMVEMLCFGE EMLCFGE(FASBPRINT)
```

This command removes the emulation configuration entry FASBPRINT from the configuration file.

[Top](#)

---

### Error messages

#### \*ESCAPE Messages

##### CPF85FA

Emulation entry QEMDFTCFGE was not removed.

##### CPF853E

Emulation configuration entry &1 not found.

##### CPF854B

Internal error in emulation configuration routines.

[Top](#)



---

## Remove Environment Variable (RMVENVVAR)

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

Parameters  
Examples  
Error messages

The Remove Environment Variable (RMVENVVAR) command removes an environment variable (or all environment variables) that exists.

**Restriction:** You must have \*JOBCTL special authority to use this command to remove system-level environment variables.

Top

---

### Parameters

Keyword	Description	Choices	Notes
ENVVAR	Environment variable	<i>Character value</i> , *ALL	Required, Positional 1
LEVEL	Level	* <u>J</u> OB, *SYS	Optional

Top

---

### Environment variable (ENVVAR)

Specifies the name of the environment variable to be removed. If an environment variable by this name does not exist at the specified level (LEVEL parameter), error message CPFA981 is issued.

If the special value of \*ALL is specified, then all environment variables at the specified level (LEVEL parameter) are removed.

**Note:** The case is preserved when lowercase characters are specified.

If an apostrophe is intended, two apostrophes must be used (").

This is a required parameter.

Top

---

### Level (LEVEL)

Specifies the level of the environment variable.

The possible values are:

\*JOB This is a job-level environment variable.

\*SYS This is a system-level environment variable.

Top

---

## Examples

### Example 1: Remove a Job-level Environment Variable

```
RMVENVVAR  ENVVAR(altdir)
```

This command removes the job-level environment variable named altdir.

### Example 2: Remove a System-level Environment Variable

```
RMVENVVAR  ENVVAR(home)  TYPE(*SYS)
```

This command removes the system-level environment variable named home.

### Example 3: Remove All Job-level Environment Variables

```
RMVENVVAR  ENVVAR(*ALL)
```

This commands removes all job-level environment variables.

### Example 4: Remove All System-level Environment Variables

```
RMVENVVAR  ENVVAR(*ALL)  LEVEL(*SYS)
```

This commands removes all system-level environment variables.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPFA981

Environment variable does not exist.

#### CPFA98E

\*JOBCTL special authority required to update system-level environment variables.

[Top](#)

---

## Remove EWC Barcode Entry (RMVEWCBCDE)

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

Parameters  
Examples  
Error messages

The Remove Extended Wireless Controller Bar Code Entry (RMVEWCBCDE) command removes the bar code entry for the specified bar code group.

**Restriction:** To execute this command, the user profile requires \*IOSYSCFG special authority.

Top

---

### Parameters

Keyword	Description	Choices	Notes
BCDGRP	Barcode group	<i>Name</i>	Required, Positional 1
INZMBR	Initialization source member	<i>Name</i>	Required, Positional 2
INZFILE	Initialization source file	<i>Qualified object name</i>	Optional, Positional 3
	Qualifier 1: Initialization source file	<i>Name</i> , <u>QEWC SRC</u>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u>	

Top

---

### Barcode group (BCDGRP)

Specifies the name of the bar code group entry that is being removed.

Top

---

### Initialization source member (INZMBR)

Specifies the name of the source file member containing the bar code entry that is being removed. The bar code entry contains the extended wireless controller configuration data.

Top

---

### Initialization source file (INZFILE)

Specifies the name of the source physical file that contains the source file member.

The name of the source file can be qualified by one of the following library values:

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

**\*CURLIB**

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

*library-name*

Specify the name of the library to be searched.

The possible values are:

QEWCSRC

The source file name QEWCSRC is used.

*source-file-name*

Specify the name of the source physical file that contains the source member.

Top

---

## Examples

```
RMVEWCBCDE BCDGRP(BCD01) INZMBR(EWC01) INZFILE(QGPL/QEWCSRC)
```

This command removes the bar code entry for bar code group BCD01 in source file member EWC01 in source file QEWCSRC in library QGPL.

Top

---

## Error messages

None

Top

---

## Remove EWC PTC Entry (RMVEWCPTCE)

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

Parameters  
Examples  
Error messages

The Remove Extended Wireless Controller PTC Entry (RMVEWCPTCE) command removes the Portable Transaction Computer (PTC) entry for the specified PTC group.

**Restriction:** To execute this command, the user profile requires \*IOSYSCFG special authority.

Top

---

### Parameters

Keyword	Description	Choices	Notes
PTCGRP	PTC group	<i>Name</i>	Required, Positional 1
INZMBR	Initialization source member	<i>Name</i>	Required, Positional 2
INZFILE	Initialization source file	<i>Qualified object name</i>	Optional, Positional 3
	Qualifier 1: Initialization source file	<i>Name</i> , <u>QEWC SRC</u>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u>	

Top

---

### PTC group (PTCGRP)

Specifies the PTC group name of the entry being removed.

Top

---

### Initialization source member (INZMBR)

Specifies the name of the source file member containing the PTC entry that is being removed. The source file member contains extended wireless controller configuration data.

Top

---

### Initialization source file (INZFILE)

Specifies the name of the source physical file that contains the source file member.

The name of the source file can be qualified by one of the following library values:

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

**\*CURLIB**

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

*library-name*

Specify the name of the library to be searched.

The possible values are:

QEWCSRC

The source file name QEWCSRC is used.

*source-file-name*

Specify the name of the source physical file that contains the source member.

Top

---

## Examples

```
RMVEWCPTCE PTCGRP(PTC01) INZMBR(EWC01) INZFILE(QGPL/QEWCSRC)
```

This command removes the PTC entry for PTC group PTC01 in source file member EWC01 in source file QEWCSRC in library QGPL.

Top

---

## Error messages

None

Top



---

## Remove Exit Program (RMVEXITPGM)

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

Parameters  
Examples  
Error messages

The Remove Exit Program (RMVEXITPGM) command removes an exit program entry for a specific exit point that is registered or unregistered. An unregistered exit point is an exit point that the registration facility created in the absence of an exit point at the time an exit program was added.

[Top](#)

---

### Parameters

Keyword	Description	Choices	Notes
EXITPNT	Exit point	<i>Simple name</i>	Required, Positional 1
FORMAT	Exit point format	<i>Simple name</i>	Required, Positional 2
PGMNBR	Program number	1-2147483647, *ALL	Required, Positional 3

[Top](#)

---

### Exit point (EXITPNT)

Specifies the name of an existing exit point for which the exit program is being removed.

[Top](#)

---

### Exit point format (FORMAT)

Specifies the name of the exit point format of the exit program that is being removed.

[Top](#)

---

### Program number (PGMNBR)

Specifies the program number of the exit program being removed.

The possible values are:

**\*ALL** Remove all exit programs for the specified exit point format (FORMAT parameter) from the specified exit point (EXITPNT parameter).

*program-number*

Remove the exit program associated with the specified program number.

[Top](#)

---

## Examples

```
RMVEXITPGM  EXITPNT(USER_EXIT_ONE)  FORMAT(EXIT1)  PGMNBR(1)
```

This command removes the exit program for exit point USER\_EXIT\_ONE that was added with program sequence number 1 for exit point format EXIT1.

[Top](#)

---

## Error messages

None

[Top](#)

---

## Remove Font Table Entry (RMVFNTTBLE)

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

Parameters  
Examples  
Error messages

The Remove Font Table Entry (RMVFNTTBLE) command removes an entry in the specified font table. This command removes an entry in the user font mapping tables used by Print Services Facility (PSF) that controls:

1. Host-resident to printer-resident font character set mapping
2. Printer-resident to host-resident font character set mapping
3. Host-resident to printer-resident code page mapping
4. Printer-resident to host-resident code page mapping
5. Printer-resident to printer-resident font substitution mapping

The entry must have previously been added to the user tables by running the Add Font Table Entry (ADDFNTTBLE) command.

In performing the printer to host and host to printer font mapping (first four tables listed above), the user tables are searched first for a match. If no match is found, then the system font or code page tables are searched.

For the printer-resident to printer-resident font substitution table, the following processing is done by the system:

- If the printer-resident font specified in the print job is supported by the printer, then it is used. The printer-resident to printer-resident font substitution table is not searched.
- If the printer-resident font specified in the print job is not supported by the printer, then the printer-resident to printer-resident font substitution table is searched.
  - If a matching entry is found in the printer-resident font substitution table and the entry is supported by the printer, then the specified substitute font in the printer-resident font substitution table is used.
  - If a matching entry is not found in the printer-resident font substitution or if the entry is not supported by the printer, then the system will use its internal font substitution tables to perform font substitution.

Refer to Printer Device Programming, SC41-5713 for more information on font mapping tables.

### Restrictions:

- The PSF feature is required to use this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
FNTTBL	Font table	Single values: *PHFCS, *HPFCS, *PHCP, *HPCP Other values: <i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Font table	<i>Name</i>	
	Qualifier 2:	<i>Name</i> , *CURLIB, *LIBL	

Keyword	Description	Choices	Notes
PHFCS	Printer font entry	<i>Element list</i>	Optional
	Element 1: Identifier	1-65535	
	Element 2: Width	1-32767, *NONE, *PTSIZE	
	Element 3: Attributes	*NONE, *BOLD, *ITALIC, *BOLDITC, *DBLWIDE, *ITCDBLWIDE	
	Element 4: Graphic character set	<i>Integer</i> , *SYSVAL	
	Element 5: Point size	1.0-999.9, *WIDTH, *NONE	
PHCP	Printer code page entry	<i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i> , *SYSVAL	
	Element 2: Code page	<i>Integer</i>	
HPFCS	Host font entry	<i>Element list</i>	Optional
	Element 1: Font character set	<i>Name</i>	
	Element 2: Type	*RASTER, *OUTLINE	
HPCP	Host code page entry	<i>Element list</i>	Optional
	Element 1: Code page	<i>Name</i>	
PPFCS	Printer to printer font	<i>Element list</i>	Optional
	Element 1: Identifier	1-65535	
	Element 2: Point size	1.0-999.9, *ALL, *NONE	

Top

## Font table (FNTTBL)

Specifies the font table to be removed.

### Single values

#### \*PHFCS

The printer-resident to host-resident font character set table is to be removed.

#### \*PHCP

The printer-resident to host-resident code page mapping table is to be removed.

#### \*HPFCS

The host-resident to printer-resident font character set table is to be removed.

#### \*HPCP

The host-resident to printer-resident code page mapping table is to be removed.

### Qualifier 1: Font table

*name* Specify the name of the font table to be removed.

### Qualifier 2: Library

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

#### \*CURLIB

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

*name* Specify the name of the library where the font table is located.

---

## Printer to host font (PHFCS)

Specifies the printer-resident font entry to be removed.

### Element 1: Printer font

#### Element 1: Identifier

##### **1-65535**

Specify the printer-resident font identifier to be mapped to a host-resident font.

#### Element 2: Width

##### **\*NONE**

No width is specified for this font identifier. \*NONE should be specified when mapping to an outline font.

##### **\*PTSIZE**

The width for this font identifier will be calculated from the point size specified. When \*PTSIZE is specified for width, the point size parameter cannot be \*NONE or \*WIDTH. When mapping a typographic raster font (2304 - 3839, 4096 - 53247, 61440 - 65534), a point size value should be specified. The width value can be \*PTSIZE or a value can be given.

##### **1-32767**

Specify a width for the font identifier. When mapping a fixed pitch raster font (1 - 750, 3840 - 4095), a width should be specified. The point size value can be \*WIDTH or a value can be given. Refer to Printer Device Programming, SC41-5713 for more information on font widths for printer-resident fonts.

#### Element 3: Attributes

##### **\*NONE**

No special font attributes are specified on this font.

##### **\*BOLD**

The printer-resident font is a bold font.

##### **\*ITALIC**

The printer-resident font is an italic font.

##### **\*BOLDITC**

The printer-resident font is a bold italic font.

##### **\*DBLWIDE**

The printer-resident font is a double wide font.

##### **\*ITCDBLWIDE**

The printer-resident font is an italic double wide font.

#### Element 4: Graphic character set

##### \*SYSVAL

The graphic character set specified in the system value QCHRID is used.

##### *graphic-character-identifier*

Specify the graphic character set for the font. The graphic character set is the first part of the graphic character identifier which consists of the graphic character set and code page.

#### Element 5: Point size

##### \*WIDTH

The font point size is computed from the font width value specified. When mapping a fixed pitch raster font (1 - 750, 3840 - 4095), it is recommended that a width value should be specified and the point size value should be \*WIDTH.

##### \*NONE

No point size is specified for this font identifier. \*NONE should be specified when mapping to an outline font.

##### *1.0-999.9*

Specify a point size ranging from 1.0 through 999.9. When mapping a typographic raster font (2304 - 3839, 4096 - 53247, 61440 - 65534), a point size value should be specified.

Top

---

## Printer to host code page (PHCP)

Specifies the printer-resident code page entry to be removed.

### Element 1: Printer code page

#### Element 1: Graphic character set

##### \*SYSVAL

The graphic character set specified in the system value QCHRID is used. A change to this system value will only take effect for the font mapping tables when the print writer is started. If QCHRID is changed and a printer is currently active, you must end the print writer and start it again.

##### *integer-number*

Specify the graphic character set for the printer-resident code page. The graphic character set is the first part of the graphic character identifier which consists of the graphic character set and code page. For example, for the graphic character identifier 697 500, 697 is the graphic character set and 500 is the code page. In this example, specify 697 for the graphic character set.

#### Element 2: Code page

##### *integer-number*

Specify the printer-resident code page value.

---

## Host to printer font (HPFCS)

Specifies the host-resident font entry to be removed.

### Element 1: Host font

#### Element 1: Font character set

*name* Specify the font character set.

#### Element 2: Type

##### \*RASTER

The host-resident font is a raster font.

##### \*OUTLINE

The host-resident font is an outline font.

---

## Host to printer code page (HPCP)

Specifies the host-resident code page entry to be removed.

### Element 1: Host code page

#### Element 1: Code page

*name* Specify the name of the host-resident code page.

---

## Printer to printer font (PPFCS)

Specifies the printer-resident font substitution mapping to be removed.

### Element 1: From printer font

#### Element 1: Identifier

*1-65535*

Specify the printer-resident font identifier for which the substitution printer-resident font is to be changed. Refer to Printer Device Programming, SC41-5713 for more information

on printer-resident fonts that are supported, and which ones are scalable (require point size) and which ones are not scalable (specify point size \*NONE).

## Element 2: Point size

### \*NONE

No font point size is specified. This should be specified for all non-scalable fonts.

**\*ALL** Specifies that all point sizes for an outline (scalable) font will be mapped. If the font is not scalable, then this will be treated the same as \*NONE.

### 1.0-999.9

Specify a point size ranging from 1.0 through 999.9. When mapping a typographic raster font (2304 - 3839, 4096 - 53247, 61440 - 65534), a point size value should be specified.

Top

---

## Examples

### Example 1: Remove Font Entry

```
RMVFNTTBLE FNTTBL(*PHFCS) PHFCS(254 84 *NONE 2039 7.0)
```

This command removes an entry from the QPHFCS table (printer resident to host-resident font character set table).

### Example 2: Remove Code Page Entry

```
RMVFNTTBLE FNTTBL(*PHCP) PHCP(*SYSVAL 38)
```

This command removes an entry from the QPHCP table (printer resident to host-resident code page table).

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2182

Not authorized to library &1.

#### CPF2283

Authorization list &1 does not exist.

#### CPF88D2

Font table &1 in library &2 not changed.

#### CPF9810

Library &1 not found.

#### CPF9820

Not authorized to use library &1.

#### CPF9822

Not authorized to file &1 in library &2.







---

## Remove Filter Action Entry (RMVFTRACNE)

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

Parameters  
Examples  
Error messages

The Remove Filter Action Entry (RMVFTRACNE) command allows the user to remove an action entry from the specified filter object.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FILTER	Filter	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Filter	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
GROUP	Group	<i>Name</i>	Required, Positional 2

Top

---

### Filter (FILTER)

Specifies the qualified name of the filter from which the action entry is being removed.

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 filter. 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 filter is located.

*filter-name*

Specify the name of the filter.

Top

---

### Group (GROUP)

Specifies the group that identifies the action entry being removed.

Top

---

### Examples

```
RMVFTRACNE FILTER(MYLIB/MYFILTER) GROUP(CHICAGO)
```

This command removes the action entry identified by the group CHICAGO in the filter MYFILTER in library MYLIB.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2150

Object information function failed.

#### CPF2151

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

#### CPF812F

Filter damaged.

#### CPF91DD

Action entry for group &4 not found.

#### CPF91EC

Internal processing error occurred.

#### CPF91E8

Internal processing error occurred.

#### CPF9802

Not authorized to object &2 in &3.

#### CPF9803

Cannot allocate object &2 in library &3.

#### CPF9807

One or more libraries in library list deleted.

#### CPF9808

Cannot allocate one or more libraries on library list.

#### CPF9830

Cannot assign library &1.

Top

---

## Remove Filter Selection Entry (RMVFTRSLTE)

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

Parameters  
Examples  
Error messages

The Remove Filter Selection Entry (RMVFTRSLTE) command allows you to remove a selection entry from a filter object.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FILTER	Filter	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Filter	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SEQNBR	Sequence number	1-9999	Required, Positional 2

Top

---

### Filter (FILTER)

Specifies the qualified name of the filter from which the selection entry is being removed.

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 filter. 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 filter is located.

*filter-name*

Specify the name of the filter that is used.

Top

---

### Sequence number (SEQNBR)

Specifies the sequence number of the selection entry to be removed. Selection entries in a filter are numbered in sequence. When a filter is applied, the selection entries with the lower sequence numbers are evaluated first. Specify a number from 1 through 9999.

Top

---

## Examples

```
RMVFTRSLTE  FILTER(MYLIB/MYFILTER)  SEQNBR(10)
```

This command removes selection entry 0010 from filter MYFILTER in library MYLIB.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2150

Object information function failed.

#### CPF2151

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

#### CPF812F

Filter damaged.

#### CPF91DC

Selection entry with sequence number &4 not found.

#### CPF91EC

Internal processing error occurred.

#### CPF91E8

Internal processing error occurred.

#### CPF9802

Not authorized to object &2 in &3.

#### CPF9803

Cannot allocate object &2 in library &3.

#### CPF9807

One or more libraries in library list deleted.

#### CPF9808

Cannot allocate one or more libraries on library list.

#### CPF9830

Cannot assign library &1.

Top

---

## Remove ICF Device Entry (RMVICFDEVE)

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

Parameters  
Examples  
Error messages

The Remove Intersystem Communications Function Device Entry (RMVICFDEVE) command removes one or more program device entries from the specified ICF file.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FILE	ICF communication file	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: ICF communication file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
PGMDEV	Program device	Values (up to 50 repetitions): <i>Character value</i>	Required, Positional 2

Top

---

### ICF communication file (FILE)

Specifies the ICF file from which the program device entries are to be removed.

This is a required parameter.

#### Qualifier 1: ICF communication file

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

*name* Specify the library where the ICF file is located.

Top

---

### Program device (PGMDEV)

Specifies the program device entries to be removed from the ICF file.

You can specify 50 values for this parameter.

This is a required parameter.

*character-value*

Specify the name of the program device entry to be removed.

Top

---

## Examples

```
RMVICFDEVE FILE(ICFHIST) PGMDEV (CHICAGO NEWYORK DENVER)
```

This command removes the program devices of CHICAGO, NEWYORK, and DENVER from the ICF file ICFHIST.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF7366

Devices not removed from file &1 in &2.

Top



---

## Remove Image Catalog Entry (RMVIMGCLGE)

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

Parameters  
Examples  
Error messages

The Remove Image Catalog Entry (RMVIMGCLGE) command removes an image catalog entry from an image catalog object (\*IMGCLG) in library QUSRSYS. Optionally, the image files associated with the catalog entry can be deleted from the system.

If the image catalog is currently loaded into a virtual device, an error message is sent and the image is not removed from the image catalog. The Load/Unload Image Catalog (LODIMGCLG) command can be used to unload the current image catalog from the virtual device.

### Restrictions:

- This command is shipped with public \*EXCLUDE authority.
- The following authorities are required to remove an image catalog entry:
  1. Execute (\*EXECUTE) authority to library QUSRSYS.
  2. \*CHANGE authority to the image catalog.
  3. Execute (\*X) authority to each directory in the image catalog path name.
- This command is not supported for reference and dependent image catalogs.

Top

---

## Parameters

Keyword	Description	Choices	Notes
IMGCLG	Image catalog	<i>Name</i>	Required, Positional 1
IMGCLGIDX	Image catalog index	1-256, *VOL	Required, Positional 2
VOL	Image catalog volume	<i>Character value</i>	Optional
KEEP	Keep image file	*YES, *NO	Optional

Top

---

## Image catalog (IMGCLG)

Specify the image catalog that the image is to be removed from.

This is a required parameter.

*name* Specify the name of the image catalog.

Top

---

## Image catalog index (IMGCLGIDX)

Specifies the image catalog index of the entry to be removed.

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

**\*VOL** Specifies to remove the entry by specifying a volume name. This value is only allowed for tape type image catalogs.

Top

---

## Image catalog volume (VOL)

Specifies the volume name of the entry to be removed.

**name** Specify the volume name of the entry to be removed. For optical image catalogs, the first volume that matches the volume specified will be removed.

Top

---

## Keep image files (KEEP)

Specifies whether the image associated with the image catalog entry is to be kept.

**\*YES** Specify that the image associated with the image catalog entry will not be deleted from the system.

**\*NO** Specify that the image file associated with the image catalog entry will be deleted from the system.

Top

---

## Examples

### Example 1: Removing an Image Catalog Entry

```
RMVIMGCLGE  IMGCLG(MYCLG)  IMGCLGIDX(2)
```

This command removes the image catalog entry at index **2** from image catalog **MYCLG**. The associated image is not deleted.

### Example 2: Remove an Image Catalog Entry and Delete the Image File

```
RMVIMGCLGE  IMGCLG(MYCLG)  IMGCLGIDX(1)  KEEP(*NO)
```

This command removes the image catalog entry at index **1** from image catalog **MYCLG**. The associated image file is deleted from the system.

### Example 3: Removing a Tape Image Catalog Entry by Volume Name

```
RMVIMGCLGE  IMGCLG(TAPECLG)  IMGCLGIDX(*VOL)  VOL(TAP001)  
KEEP(*NO)
```

This command removes the tape image catalog entry with a volume name of **TAP001** from image catalog **TAPECLG**. The associated image file is deleted from the system.

Top

---

## Error messages

### \*ESCAPE Messages

**CPFBC30**

Catalog entry at index &2 not removed from image catalog &1.

**CPFBC31**

Catalog entry at index &2 not removed from image catalog &1.

**CPFBC45**

Image catalog &1 not found.

**CPF9802**

Not authorized to object &2 in &3.

**CPF9820**

Not authorized to use library &1.

Top



---

## Remove IP over SNA Interface (RMVIPSIFC)

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

Parameters  
Examples  
Error messages

The Remove IP over SNA Interface (RMVIPSIFC) command removes an AF\_INET sockets over SNA interface (an IP address by which this local host is known on the SNA transport). This command can be used to remove interfaces that have been specified with the Add IP over SNA Interface (ADDIPSIFC) CL command. The interface to be removed is identified by its internet address (INTNETADR).

### Restrictions:

1. The user must have \*IOSYSCFG authority to use this command.
2. The interface cannot be active when you submit this command. Use the End IP over SNA Interface (ENDIPSIFC) CL command to deactivate the interface.
3. There can be no configured routes whose NEXTHOP internet address can be reached only through the network associated with the IP over SNA interface to be removed.

Top

---

## Parameters

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

Top

---

## Internet address (INTNETADR)

Specifies the internet address of an interface that had previously been added to the SNA configuration with the Add IP over SNA Interface (ADDTCPIFC) 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. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

This is a required parameter.

### *internet-address*

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

Top

---

## Examples

```
RMVIPSIFC '9.5.1.248'
```

This command removes the interface with the IP address 9.5.1.248.

---

## Error messages

### \*ESCAPE Messages

#### TCP8050

\*IOSYSCFG authority required to use &1.

#### TCP9999

Internal system error in program &1.

---

## Remove IP over SNA Location (RMVIPSLOC)

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

Parameters  
Examples  
Error messages

The Remove IP over SNA Location Entry (RMVIPSLOC) command removes an AF\_INET sockets over SNA location entry. This command can be used to remove location entries that have been specified with the Add IP over SNA Location Entry (ADDIPSLOC) CL command. The location entry to be removed is identified by its remote route destination (RMTDEST) and subnet mask (SUBNETMASK).

**Restriction:** The user must have \*IOSYSCFG authority to use this command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
RMTDEST	Remote destination	<i>Character value</i>	Required, Positional 1
SUBNETMASK	Subnet mask	<i>Character value, *HOST</i>	Required, Positional 2

Top

---

### Remote destination (RMTDEST)

Specifies the remote route destination of the location entry to be removed. The remote route destination is specified in the form *nnn.nnn.nnn.nnn* where *nnn* is a decimal number ranging from 0 through 255. If the remote route destination address is entered from a command line, the address must be enclosed in apostrophes.

This is a required parameter.

Top

---

### Subnet mask (SUBNETMASK)

Specifies the subnet mask of the location entry being removed.

This is a required parameter.

The possible values are:

#### **\*HOST**

Specify this value when the internet address value specified in the remote route destination field is a host address. The subnet mask value is calculated to be 255.255.255.255.

### *subnetwork-mask*

Specify the subnet mask in the forms *nnn.nnn.nnn.nnn* where *nnn* is a decimal number ranging from 0 through 255. If the subnet mask address is entered from a command line, the address must be enclosed in apostrophes.

Top

---

## Examples

### Example 1: Removing a Location Entry for a Subnetwork

```
RMVIPSLOC RMTDEST('128.2.0.0') SUBNETMASK('255.255.255.128')
```

This command removes a location entry for a subnetwork with network 128.2 and subnet mask of 255.255.255.128.

### Example 2: Removing a Location Entry for a Network

```
RMVIPSLOC RMTDEST(128.3.0.0) SUBNETMASK('255.255.0.0')
```

This command removes a location entry for network 128.3.

Top

---

## Error messages

None

Top



---

## Remove IP over SNA Route (RMVIPS RTE)

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

Parameters  
Examples  
Error messages

The Remove IP over SNA Route Entry (RMVIPS RTE) command removes an AF\_INET sockets over SNA route. This command can be used to remove routes that have been specified with the Add IP over SNA Route (ADDIPS RTE) CL command. The route to be removed is identified by its route destination (RTEDEST), subnet mask (SUBNETMASK), and next hop (NEXTHOP).

**Restriction:** The user must have \*IOSYSCFG authority to use this command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
RTEDEST	Route destination	Character value	Required, Positional 1
SUBNETMASK	Subnet mask	Character value, *HOST	Required, Positional 2
NEXTHOP	Next hop	Character value	Required, Positional 3

Top

---

### Route destination (RTEDEST)

Specifies the route destination of the route to be removed. The route destination is specified in the form *nnn.nnn.nnn.nnn* where *nnn* is a decimal number ranging from 0 to 255. If the route destination address is entered from a command line, the address must be enclosed in apostrophes.

This is a required parameter.

Top

---

### Subnet mask (SUBNETMASK)

Specifies the subnet mask of the route to be removed.

This is a required parameter.

**Note:** The network portion must be equal to one bits in the subnetmask. The host portion and the subnetmask portion of an address must both be at least two bits wide.

The possible values are:

#### \*HOST

Specify this value when the internet address value specified in the route destination field is a host address.

### *subnet-mask*

Specify the subnet mask in the form *nnn.nnn.nnn.nnn* where *nnn* is a decimal number ranging from 0 to 255. If the subnet mask address is entered from a command line, the address must be enclosed in apostrophes.

Top

---

## Next hop (NEXTHOP)

Specifies the next hop of the route to be removed. The next hop is specified in the form *nnn.nnn.nnn.nnn* where *nnn* is a decimal number ranging from 0 to 255. If the next hop address is entered from a command line, the address must be enclosed in apostrophes.

This is a required parameter.

Top

---

## Examples

```
RMVIPS RTEDEST('128.2.0.0') SUBNETMASK('255.255.255.128')
      NEXTHOP ('128.3.4.5')
```

This command removes a network route entry for a subnetwork with network 128.2 and subnet mask of 255.255.255.128.

Top

---

## Error messages

### \*ESCAPE Messages

#### TCP264A

&2 &1 not removed.

#### TCP8050

\*IOSYSCFG authority required to use &1.

#### TCP9999

Internal system error in program &1.

Top

---

## Remove Job Queue Entry (RMVJOBQE)

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

Parameters  
Examples  
Error messages

The Remove Job Queue Entry (RMVJOBQE) command removes a job queue entry from the specified subsystem description. Jobs on the job queue remain on the queue when the job queue entry is removed from the subsystem description.

### Restrictions:

1. To use this command, you must have:
  - object operational (\*OBJOPR), object management (\*OBJMGT), and read (\*READ) authority to the specified subsystem description and execute (\*EXECUTE) authority to the library containing the subsystem description.
2. A job queue entry cannot be removed if any currently active jobs were started from the job queue.

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>	
JOBQ	Job queue	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Job queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Top

---

## Subsystem description (SBSD)

Specifies the name and library of the subsystem description from which the job queue entry is removed.

This is a required parameter.

### Qualifier 1: Subsystem description

*name* Specify the name of the subsystem description from which the job queue entry is removed.

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

- QSYSSBSD

### Qualifier 2: Library

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

### **\*CURLIB**

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

*name* Specify the library where the subsystem description is located.

Top

---

## **Job queue entry job queue (JOBQ)**

Specifies the name and library of the job queue that has its job queue entry removed from the subsystem description.

This is a required parameter.

### **Qualifier 1: Job queue**

*name* Specifies the name of the job queue for the entry that is being removed.

### **Qualifier 2: Library**

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

### **\*CURLIB**

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

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

Top

---

## **Examples**

```
RMVJOBQE  SBSDB(MYLIB/NIGHTRUN)  JOBQ(MYLIB/BATCH2)
```

This command removes the job queue entry that refers to the BATCH2 job queue in MYLIB from the NIGHTRUN subsystem description stored in library MYLIB.

Top

---

## **Error messages**

### **\*ESCAPE Messages**

#### **CPF1619**

Subsystem description &1 in library &2 damaged.

#### **CPF1691**

Active subsystem description may or may not have changed.

#### **CPF1697**

Subsystem description &1 not changed.

Top

---

## Remove Job Schedule Entry (RMVJOBSCDE)

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

Parameters  
Examples  
Error messages

The Remove Job Schedule Entry (RMVJOBSCDE) command allows you to remove an entry, entries, or generic entries in the job schedule. Each job schedule entry corresponds to one batch job, and contains the information needed to automatically run the job once or at regularly scheduled intervals. A message is sent to you and the message queue specified in the job schedule entry when an entry is successfully removed.

### Restrictions:

1. To remove entries, you must be running under a user profile which has job control (\*JOBCTL) special authority; otherwise you can remove only those entries that you added.
2. To use this command, you must have:
  - Use (\*USE) authority to object QDFTJOBSCD, type \*JOBSCD, in library QUSRSYS and execute (\*EXECUTE) authority to library QUSRSYS.

Top

---

## Parameters

Keyword	Description	Choices	Notes
JOB	Job name	Generic name, name	Required, Positional 1
ENTRYNBR	Entry number	000001-999999, *ONLY, *ALL	Optional

Top

---

## Job name (JOB)

Specifies the name of the job schedule entry.

This is a required parameter.

### *generic-name*

Specify the generic name of the job schedule entry. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. If a generic name is specified, then all entries with names that begin with the generic name, and for which the user has authority, are removed. If a generic name is specified, ENTRYNBR(\*ALL) must also be specified. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete job name.

*name* Specify the name of the job schedule entry that you want to remove.

Top

---

## Entry number (ENTRYNBR)

Specifies the number of the job schedule entry you want to remove. The message sent when an entry is successfully added contains the entry number. You can also determine the entry number by using the Work with Job Schedule Entries (WRKJOBSCDE) command. Press F11 from the WRKJOBSCDE display to show the entry numbers of the selected entries.

### \*ONLY

Only one entry in the job schedule has the job name specified for the JOB parameter. If \*ONLY is specified and more than one entry has the specified job name, no entries are removed and an error message is sent.

**\*ALL** All entries with the specified job name are removed.

### **000001-999999**

Specify the number of the job schedule entry you want to remove.

Top

---

## Examples

### Example 1: Removing Job Schedule Entries

```
RMVJOBSCDE JOB(SAMPLE*) ENTRYNBR(*ALL)
```

This command removes all the job schedule entries whose job names start with SAMPLE.

### Example 2: Removing an Individual Job Schedule Entry

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

This command removes the job PAYROLL in the job schedule.

### Example 3: Removing a Generic Job Schedule Entry

```
RMVJOBSCDE JOB(PAY*) ENTRYNBR(*ALL)
```

This command removes all entries in the job schedule that have the prefix PAY in their names.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1628

Job schedule entry &3 number &4 not found.

#### CPF1629

Not authorized to job schedule &1.

#### CPF1630

Not authorized to job schedule entry &3 number &4.

#### CPF1631

&3 entries successfully removed, &4 entries not removed.

#### CPF1636

More than one entry with specified entry job name found.

#### CPF1637

Job schedule &1 in library &2 in use.

**CPF1638**

Job schedule entry &3 number &4 in use.

**CPF1640**

Job schedule &1 in library &2 does not exist.

**CPF1641**

Job schedule &1 in library &2 damaged.

**CPF1645**

No job schedule entries found for specified name.

**CPF1646**

Entry number must be \*ALL when generic name specified.

[Top](#)





---

## Remove Journalled Changes (RMVJRNCHG)

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

Parameters  
Examples  
Error messages

The Remove Journalled Changes (RMVJRNCHG) command removes the changes that have been journalled for a particular journalled object. The journalled changes are removed from the object from the specified starting point to the ending point. The journal entries are processed in reverse of the order in which they were placed into the journal receiver, from the most recent to the oldest. The starting point can be identified as the last journal entry in the specified journal receiver range, the point at which an object was last saved, or a particular entry in the receiver range. The ending point can be the first journal entry or a particular entry in the specified journal receiver range, or the point at which an object was opened by a specified job. The CMTBDY parameter can be used for handling changes that are pending.

**Note:** No database object level changes are processed by RMVJRNCHG.

**Note:** The Display Journal (DSPJRN) command can be used to help determine the desired starting and ending points.

A list of journalled objects may be specified. The journalled changes for all objects are removed in the order that the journal entries are found on the journal (the reverse of the order that the changes were originally made to the objects).

If a journal code J entry type SX (Exit JRNSTATE(\*STANDBY)) entry is found, the operation ends for all objects specified regardless of the OBJERROPT value specified. Objects may be only partially updated from the journal entries.

Additionally, the command can end for an individual object when journal entries list operations which cannot be replayed by the command. If this happens, the operation will continue for the other objects specified if OBJERROPT(\*CONTINUE) is used. For example, the command ends for an object when a journal entry is found that indicates one of the following has occurred:

- A physical database file is cleared
- A physical database file member is saved and its storage is freed
- Journal initial program load (IPL) synchronization fails
- The system has already applied or removed the changes through the Apply Journal Changes (APYJRNCHG) command or the RMVJRNCHG command.

See the Journal Management information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter> for a complete listing of the various entries and how they are handled by this command including those entries which can stop the command.

The command also ends for an object on illogical conditions. If the command ends due to illogical conditions and it is logically possible to restart the operation, you can issue the command again for that object, specifying a new starting sequence number. Be aware though, that most illogical conditions are the result of starting the remove at the wrong journal entry.

Journal entry changes can be removed even if the sequence numbers have been reset. The system handles this condition, sends an informational message, and continues the removal of journalled changes.

### Restrictions:

- This command is shipped with public \*EXCLUDE authority and the QPGMR and QSRV user profiles have private authorities to use the command.
- The objects specified on this command must currently be having their changes journaled, and they must have been journaled to the specified journal throughout the period indicated on the command.
- Before images are required for the objects. See the appropriate start journaling command.
- The objects indicated on the command are allocated exclusively while the changes are being removed. If an object cannot be allocated exclusively, the command ends and no journaled changes are removed from the objects.
- If there is no journal entry that represents the entry specified on the FROMENTLRG, FROMENT, TOENTLRG, or TOENT parameter, the command ends and no journaled changes are removed from the objects.
- If the journal sequence numbers have been reset in the range of the receivers specified, and a sequence number is specified on the FROMENTLRG, FROMENT, TOENTLRG, or TOENT parameter, the first occurrence of the sequence number specified on either parameter is used.
- The TOJOB0 parameter cannot be used to specify when the remove journaled changes operation is to end if one or more journal receivers in the specified receiver range was attached to the journal when a receiver size option (RCVSIZOPT) or a fixed length data option (FIXLENDTA) that would have omitted this data was in effect.
- The TOJOB0 parameter cannot be used if the object for which changes are being removed from was not recording open and close entries. For further clarification, refer to the Omit journal entry (OMTJRNE) parameter for the STRJRN, STRJRNPf, and CHGJRNOBJ commands.
- This command cannot be used on or with a remote journal.
- The maximum number of objects that can have changes removed with this command is 12,000,000. If more than 12,000,000 objects are included in the specifications, an error message is sent and no changes are removed. You can change the values specified on this parameter so that the limit is not exceeded.

Top

## Parameters

Keyword	Description	Choices	Notes
JRN	Journal	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Journal	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
FILE	Journaled file identification	Values (up to 300 repetitions): <i>Element list</i>	Optional, Positional 2
	Element 1: Journaled physical file	<i>Qualified object name</i>	
	Qualifier 1: Journaled physical file	<i>Name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
	Element 2: Member	<i>Name, *ALL, *FIRST</i>	
OBJ	Objects	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
	Element 2: Object type	<i>*FILE, *DTAARA</i>	
	Element 3: Member, if data base file	<i>Name, *ALL, *FIRST</i>	

Keyword	Description	Choices	Notes
RCVRNG	Range of journal receivers	Single values: <b>*CURRENT</b> , <b>*LASTSAVE</b> Other values: <i>Element list</i>	Optional, Positional 3
	Element 1: Starting journal receiver	<i>Qualified object name</i>	
	Qualifier 1: Starting journal receiver	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b> , <b>*CURLIB</b>	
	Element 2: Ending journal receiver	<i>Qualified object name</i>	
	Qualifier 1: Ending journal receiver	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b> , <b>*CURLIB</b>	
FROMENTLRG	Starting large sequence number	Character value, <b>*LAST</b> , <b>*LASTSAVE</b>	Optional
TOENTLRG	Ending large sequence number	Character value, <b>*FIRST</b> , <b>*COMMITSTART</b>	Optional
TOJOB0	Fully qualified job name	<i>Qualified job name</i>	Optional
	Qualifier 1: Fully qualified job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
CMTBDY	Commitment boundary	<b>*YES</b> , <b>*NO</b>	Optional
OPTION	Option	<b>*NONE</b> , <b>*IGNINQMSG</b>	Optional
OBJERROPT	Object error option	<b>*CONTINUE</b> , <b>*END</b>	Optional
OUTPUT	Output	<b>*NONE</b> , <b>*OUTFILE</b>	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> , <b>*CURLIB</b>	
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> , <b>*ADD</b>	
DETAIL	Detail	<b>*ALL</b> , <b>*ERR</b>	Optional
FROMENT	Starting sequence number	1-9999999999, <b>*LAST</b> , <b>*LASTSAVE</b>	Optional
TOENT	Ending sequence number	1-9999999999, <b>*FIRST</b> , <b>*COMMITSTART</b>	Optional

Top

## Journal (JRN)

Specifies the journal that contains the journal entries being removed.

This is a required parameter.

### Qualifier 1: Journal

*journal-name*

Specify the name of the journal.

## Qualifier 2: Library

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

### \*CURLIB

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

### *library-name*

Specify the name of the library to be searched.

Top

---

## Journalized file identification (FILE)

Specifies a maximum of 300 physical database files whose journal entries are to be removed. The member in the physical file whose journalized changes are to be removed can also be specified.

Either the **Journalized file identification (FILE)** parameter or the **Objects (OBJ)** parameter must be specified, but not both.

### Element 1: Journalized physical file

#### Qualifier 1: Journalized physical file

**\*ALL** Journalized changes are removed from all physical files in the specified library whose changes have been journalized to the specified journal. The library name *must* be specified. If **\*ALL** is specified and you do not have the required authority to all of the objects, an error message is sent and the command ends.

### *file-name*

Specify the name of the database physical file whose journalized changes are to be removed.

## Qualifier 2: Library

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

### \*CURLIB

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

### *library-name*

Specify the name of the library to be searched.

### Element 2: Member

Specifies the name of the member in the file whose journalized changes are removed. If **\*ALL** is specified for the first element of this parameter, the value specified for the member name is used for all applicable files in the library. For example, if **\*FIRST** is specified for the member name, journal changes are removed from the first member of all applicable files in the library.

**\*ALL** Journalized changes are removed from all members in the file.

### \*FIRST

Journalized changes are removed from the first member in the file.

### *member-name*

Specify the name of the member in the file whose journalized changes are removed.

Top

---

## Objects (OBJ)

Specifies a maximum of 300 objects whose journal entries are being removed. The name of the member in the file whose journal entries are being removed can also be specified.

Either the **Journalized file identification (FILE)** parameter or the **Objects (OBJ)** parameter must be specified, but not both.

### Element 1: Object

#### Qualifier 1: Object

**\*ALL** All objects in the specified library of the specified type whose changes are being journaled to the specified journal have their journal entries removed. The library name must be specified. If **\*ALL** is specified and you do not have the required authority to all of the objects, an error message is sent and the command ends.

#### *object-name*

Specify the name of the object whose journal entries are being removed.

#### Qualifier 2: Library

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

#### **\*CURLIB**

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

#### *library-name*

Specify the name of the library to be searched.

### Element 2: Object type

Specifies the object type of the object that has its journal entries removed.

**\*FILE** Entries for database file members are removed.

#### **\*DTAARA**

Entries for data areas are removed.

### Element 3: Member, if data base file

Specifies the name of the member in the file that has its journal entries removed. If **\*ALL** is specified for the first element of this parameter, the value specified for the member name is used for all applicable files in the library. For example, if **\*FIRST** is specified for the member name, journal changes are removed from the first member of all applicable files in the library.

**Note:** If the specified object type is not **\*FILE**, the member name element value is ignored.

**\*ALL** Journal entries are removed from all members in the file.

#### **\*FIRST**

Journal entries are removed from the first member of the file.

#### *member-name*

Specify the name of the member in the file whose journal entries are to be removed.

Top

---

## Range of journal receivers (RCVRNG)

Specifies the starting and ending journal receivers used in removing the journaled changes. The system begins the removal operation with the starting journal receiver (specified by the first value) and proceeds through the chain of receivers until the ending receiver (specified by the last value) is processed. The values specified on the parameter represent journal receivers in reverse order from the order in which they were attached to the journal.

**Note:** The maximum number of receivers that can be included in a range of receivers is 1024. If more than 1024 receivers are included in the range specified, an error message is sent and no changes are applied. You can change the values specified on this parameter so that the limit is not exceeded.

### Single values

#### \*CURRENT

The journal receiver that is currently attached when starting to remove journal entries is used.

#### \*LASTSAVE

The range of journal receivers used is determined by the system, as a result of save information for the objects that have their recorded changes removed. This parameter value is only valid if \*LASTSAVE is also specified on the **Starting sequence number (FROMENT)** parameter, and on the **Starting large sequence number (FROMENTLRG)** parameter.

### Element 1: Starting journal receiver

#### Qualifier 1: Starting journal receiver

##### *starting-journal-receiver*

Specify the name of the journal receiver used as the first (newest) receiver whose journaled changes are to be removed.

#### Qualifier 2: Library

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

#### \*CURLIB

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

##### *library-name*

Specify the library where the journal receiver is located.

### Element 2: Ending journal receiver

#### Qualifier 1: Ending journal receiver

##### *ending-journal-receiver*

Specify the name of the journal receiver used as the last (oldest) receiver whose journaled changes are to be removed. If the end of the receiver chain is reached before a receiver of this name is found, the operation ends.

#### Qualifier 2: Library

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

#### \*CURLIB

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

##### *library-name*

Specify the library where the journal receiver is located.

---

## Starting large sequence number (FROMENTLRG)

Specifies the journal entry used as the starting point for removing file changes that were journaled.

**Note:** You can specify a value for either the **Starting sequence number (FROMENT)** parameter or the **Starting large sequence number (FROMENTLRG)** parameter, but not for both.

### \*LAST

Specifies that journaled changes are removed starting with the last journal entry in the specified receiver range.

### \*LASTSAVE

Journal entries are removed starting with the last journal entry before the last save operation. The system determines the actual starting position for each of the objects specified on the command. The parameter value implies that the object was just restored on the system.

If the restored version of the object was a version that was saved using the save-while-active function, then the system will start removing changes from the corresponding start-of-save entry whether or not this was actually the last save of the object. When using save-while-active, information needed for removing journaled changes is saved with the object and restored. When all objects specified on the remove command have been restored from save versions that used save-while-active, the system does not need to scan all the journal receivers to find the save points for the objects. This can improve the performance of the remove processing.

If the restored version of the object was a version that was saved when it was not in use (normal save), then the system also verifies that the date and time of the saved version of the object that is restored on the system is the same as the date and time that the object was last saved, as indicated on the journal.

If the dates and times do not match, no entries are removed and an inquiry message is sent to the user or system operator requesting a cancel or ignore response. If an ignore response is given to the message, the operation is attempted. A cancel response causes the operation to end, and no journal changes are removed.

If the object was last saved with the save-while-active function, the saved copy of each object includes all changes in the journal entries up to the corresponding start-of-save journal entry. In this case, the system removes changes beginning with the first journal entry preceding the start-of-save entry.

If the object was last saved when it was not in use (normal save), the saved copy of each object includes all changes in the journal entries up to the corresponding object saved journal entry. In this case, the system removes changes beginning with the first journal entry preceding the object saved entry.

**Note:** If any database file members were saved specifying \*NOCMTBDY as the second element of the SAVACTWAIT parameter on the save command and are currently in a state where apply journaled changes is required, then \*LASTSAVE must be specified.

### *starting-sequence-number*

Specify the sequence number of the first journal entry that is processed when removing journaled changes from the specified objects. The acceptable range is 1 to 18,446,744,073,709,551,600.

---

## Ending large sequence number (TOENTLRG)

Specifies the journal entry used as the ending point for removing changes that were journaled.

**Note:** You can specify a value for either the **Ending sequence number (TOENT)** parameter or the **Ending large sequence number (TOENTLRG)** parameter, but not for both.

#### **\*FIRST**

Journal entries are removed until the first entry in the specified receiver range is processed.

#### **\*COMMITSTART**

Either FROMENTLRG(\*LASTSAVE) or FROMENT(\*LASTSAVE) must be specified, along with CMTBDY(\*YES) to specify this value. Starting with the last save journal entry, only changes for journal entries for any partial transactions are removed, back to the start of the commit transaction. This option can be used when the version of the object that was restored was a version that was saved with partial transactions and Apply Journalized Changes (APYJRNCHG) is not desired.

#### *ending-sequence-number*

Specify the sequence number of the last journalized change that is removed from the specified objects. The acceptable range is 1 to 18,446,744,073,709,551,600.

Top

---

## **Fully qualified job name (TOJOB0)**

Specifies the job identifier of the job that, when it opens an object that is specified, ends the removing of journal entries by this command. For example, the specified job could be a job suspected of causing errors when the job opens a file member. The first job open entry found for any of the specified objects, is the ending point for all the objects specified.

This parameter cannot be used to remove a specific job's journal entries; all entries for all jobs are removed.

Only objects of type \*FILE have journal entries related to job opens. The TOJOB0 parameter cannot be used if the object for which changes are being removed from was not recording open and close entries. For further clarification, refer to the Omit journal entry (OMTJRNE) parameter for the STRJRN, STRJRNPF, and CHGJRNOBJ commands.

### **Other values**

#### *job-identifier*

Specify the job name, the user name, and the job number of the job to use. You can also specify that the job name only, or that the job name and the user name be used.

#### *job-name*

Specify the job name of the job.

#### *user-name*

Specify the user name of the job.

#### *job-number*

Specify the system-assigned job number.

Top



---

## Commitment boundary (CMTBDY)

Specifies whether commitment boundaries are honored when the journal entries from which journaled changes are to be removed are part of a commitment control logical unit of work (LUW). More information on the use of commitment control is in the Database information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

**Note:** For purposes of this parameter description, the TO option is used to describe either the TOENTLRG, the TOENT, or the TOJOB0 parameter, whichever is specified.

**\*YES** The journal entries are removed from the entry specified on the FROMENTLRG or FROMENT parameter to the entry indicated on the TO option, honoring commitment boundaries. The FROMENTLRG or FROMENT parameter must identify a point that is at a commitment boundary.

- If the journal entry specified on the FROMENTLRG or FROMENT parameter is in the middle of the LUW of which it is a participant, an error message is sent and the operation is not attempted.
- If the journal entry indicated on the TO option is in the middle of the LUW of which it is a participant, the operation stops at the commitment boundary before that journal entry. A diagnostic message is sent at the end of the operation.

**Note:** If a journal entry is encountered that causes the operation to end before the entry indicated on the TO option, commitment boundaries might not be honored.

**\*NO** The journal entries are removed from the entry specified on the FROMENTLRG or FROMENT parameter to the entry indicated on the TO option, regardless of commitment boundaries. Even if a journal entry within this range is a participant of the LUW, the operation is attempted.

**Note:** If CMTBDY(\*NO) is specified and any object being applied to has been restored from a saved version that contains partial transactions, the changes pending for those partial transactions will not be removed if the transactions do not complete within the specified range. The object will only be usable if all pending transactions complete within the specified range.

Top

---

## Option (OPTION)

Specifies whether additional checking should be done prior to removing journal changes.

### **\*NONE**

All protective checks are performed before any journal changes are removed.

### **\*IGNINQMSG**

Ignore inquiry message. Inquiry message CPA7050 is not presented to the user, even if the object that is being applied to is not from the last save of the object. The remove operation continues.

Top

---

## Object error option (OBJERROPT)

Specifies how the processing of journal entries should proceed when an error situation is encountered.

### **\*CONTINUE**

When a journal entry for a specific object is encountered that cannot be processed, the remaining journal entries for that object will not be processed. Processing of journal entries for other objects will continue. A diagnostic message will be sent indicating that the processing of journaled changes for that object was not successful. An indication is also placed in any output file record to indicate processing ended early for the specific object.

**\*END** When the first journal entry is encountered that cannot be successfully processed, processing will end for all objects.

Top

---

## Output (OUTPUT)

Specifies whether a list of information about the objects to whom changes were removed is created. The information can be directed to a database file.

### **\*NONE**

No data base file is created with the output. Messages are sent to the job log for the first 512 objects for which processing failed to complete successfully.

### **\*OUTFILE**

Output information about the remove operation will be directed to the database file specified on the **File to receive output (OUTFILE)** parameter.

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

Top

---

## File to receive output (OUTFILE)

Specifies the database file to which the information is directed when \*OUTFILE is specified on **Output (OUTPUT)** parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QAJRNCHG in QSYS with the format name QJOAPYRM as a model.

### **Qualifier 1: File to receive output**

#### *database-file-name*

Specify the name of the database file to which output from the command is directed. If this file does not exist, it is created in the specified library.

### **Qualifier 2: Library**

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

### **\*CURLIB**

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

#### *library-name*

Specify the name of the library to be searched.

Top

---

## Output member options (OUTMBR)

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

### **Element 1: Member to receive output**

### \*FIRST

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

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

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

### Element 2: Replace or add records

### \*REPLACE

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

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

Top

---

## Type of output information (DETAIL)

Specifies the type of information that is directed to the spooled file.

**\*ALL** The file will contain information about the command and an entry for each object that matched the object selection specified on the command.

**\*ERR** The file will contain information about the command, an entry only for each object for which the removal of journaled changes ended early.

Top

---

## Starting sequence number (FROMENT)

Specifies the journal entry used as the starting point for removing file changes that were journaled.

**Note:** You can specify a value for either the **Starting sequence number (FROMENT)** parameter or the **Starting large sequence number (FROMENTLRG)** parameter, but not for both.

### \*LAST

Specifies that journaled changes are removed starting with the last journal entry in the specified receiver range.

### **\*LASTSAVE**

Journal entries are removed starting with the last journal entry before the last save operation. The system determines the actual starting position for each of the objects specified on the command. The parameter value implies that the object was just restored on the system.

If the restored version of the object was a version that was saved using the save-while-active function, then the system will start removing changes from the corresponding start-of-save entry whether or not this was actually the last save of the object. When using save-while-active, information needed for removing journaled changes is saved with the object and restored. When all objects specified on the remove command have been restored from save versions that used save-while-active, the system does not need to scan all the journal receivers to find the save points for the objects. This can improve the performance of the remove processing.

If the restored version of the object was a version that was saved when it was not in use (normal save), then the system also verifies that the date and time of the saved version of the object that is restored on the system is the same as the date and time that the object was last saved, as indicated on the journal.

If the dates and times do not match, no entries are removed and an inquiry message is sent to the user or system operator requesting a cancel or ignore response. If an ignore response is given to the message, the operation is attempted. A cancel response causes the operation to end, and no journal changes are removed.

If the object was last saved with the save-while-active function, the saved copy of each object includes all changes in the journal entries up to the corresponding start-of-save journal entry. In this case, the system removes changes beginning with the first journal entry preceding the start-of-save entry.

If the object was last saved when it was not in use (normal save), the saved copy of each object includes all changes in the journal entries up to the corresponding object saved journal entry. In this case, the system removes changes beginning with the first journal entry preceding the object saved entry.

**Note:** If any database file members were saved specifying \*NOCMTBDY as the second element of the SAVACTWAIT parameter on the save command and are currently in a state where apply journaled changes is required, then \*LASTSAVE must be specified.

*starting-sequence-number*

Specify the sequence number of the first journal entry that is processed when removing journaled changes from the specified objects. The acceptable range is 1 to 9,999,999,999.

Top

---

## Ending sequence number (TOENT)

Specifies the journal entry used as the ending point for removing changes that were journaled.

**Note:** You can specify a value for either the **Ending sequence number (TOENT)** parameter or the **Ending large sequence number (TOENTLRG)** parameter, but not for both.

\*FIRST

Journal entries are removed until the first entry in the specified receiver range is processed.

**\*COMMITSTART**

Either FROMENTLRG(\*LASTSAV) or FROMENT(\*LASTSAV) must be specified, along with CMTBDY(\*YES) to specify this value. Starting with the last save journal entry, only changes for journal entries for any partial transactions are removed, back to the start of the commit transaction. This option can be used when the version of the object that was restored was a version that was saved with partial transactions and Apply Journaled Changes (APYJRNCHG) is not desired.

*ending-sequence-number*

Specify the sequence number of the last journaled change that is removed from the specified objects. The acceptable range is 1 to 9,999,999,999.

Top

---

## Examples

```
RMVJRNCHG  JRN(JRNA)  FILE((LIB2/PAYROLL JAN))
           RCVRNG(RCV25 RCV22)  TOENT(*FIRST)
```

This command causes the system to remove all journaled changes found in journal JRNA from member JAN of file PAYROLL in library LIB2 that are found on the journal receiver chain starting with receiver RCV25 and ending with receiver RCV22. Library search list \*LIBL is used to find journal JRNA and receivers RCV25 and RCV22.

The removal operation begins with the last journaled change on the receiver chain and ends with the first journaled change.

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPF69A9**

Internal error detected, error code &2.

#### **CPF69AA**

Unable to refer to object &1 saved with STG(\*FREE).

#### **CPF69AB**

Values for RCVRNG parameter not correct.

#### **CPF70A7**

Not all entries applied or removed for at least one object.

#### **CPF70A8**

File &1 in library &2 cannot be used.

#### **CPF70AA**

FROMENT(\*LASTSAVE) must be specified.

#### **CPF70AB**

Journal receiver &1 in library &2 not found.

#### **CPF70CC**

Cannot perform operation beyond journal entry &7.

#### **CPF70CD**

Cannot perform operation beyond journal entry &7.

#### **CPF70CE**

Cannot perform operation beyond journal entry &7.

#### **CPF70EB**

Referential constraint error on member &3.

#### **CPF70EC**

Referential constraint error. Reason code &9.

#### **CPF70EE**

Maximum encoded vector access paths for member &3.

#### **CPF7002**

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

#### **CPF7003**

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

#### **CPF7006**

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

#### **CPF7007**

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

#### **CPF701B**

Journal recovery of an interrupted operation failed.

#### **CPF704A**

Record length incorrect for member &3.

**CPF704F**  
TOJOB0 or TOJOB C parameter not valid for receiver range.

**CPF7041**  
Entry for job &3/&2/&1 not found.

**CPF7042**  
Object not journaled or journaled to different journal.

**CPF7044**  
Apply or remove of journaled entries failed, reason code &7.

**CPF7045**  
Journal receiver &1 in &2 partially damaged.

**CPF7046**  
Duplicate key not allowed for member &3.

**CPF7047**  
Member &3 file &1 in &2 full.

**CPF7048**  
Cannot perform journaled change to member &3.

**CPF7049**  
Cannot perform operation beyond journal entry &7.

**CPF7051**  
Save entry for \*&6 object not found.

**CPF7052**  
Select/omit failure in logical file over member &3.

**CPF7053**  
Values for RCVRNG parameter not correct; reason code &1.

**CPF7054**  
FROM and TO values not valid.

**CPF7056**  
Object &1 not being journaled with before images.

**CPF7057**  
\*LIBL not allowed with FILE(\*ALL) or OBJ(\*ALL).

**CPF7058**  
Apply or remove journaled entries operation failed.

**CPF705A**  
Operation failed due to remote journal.

**CPF7063**  
Cannot use file &1 as OUTFILE.

**CPF7067**  
FROMENTLRG or FROMENT option not valid. Commit boundary violation.

**CPF7068**  
Entry needed for apply or remove operation not found.

**CPF7069**  
No entries applied or removed using journal &1.

**CPF7077**  
Key mapping error on member &3.

**CPF7078**  
Cannot apply or remove changes to member &3.

**CPF9801**  
Object &2 in library &3 not found.

**CPF9802**  
Not authorized to object &2 in &3.

**CPF9803**  
Cannot allocate object &2 in library &3.

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

**CPF9810**  
Library &1 not found.

**CPF9812**  
File &1 in library &2 not found.

**CPF9820**  
Not authorized to use library &1.

**CPF9822**  
Not authorized to file &1 in library &2.

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

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

Top





---

## Remove LAN Adapter Information (RMVLANADPI)

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

Parameters  
Examples  
Error messages

The Remove Local Area Network Adapter Information (RMVLANADPI) command removes an adapter name entry from the adapter file.

Top

---

### Parameters

Keyword	Description	Choices	Notes
ADPTNAME	Adapter	Name, *ADPTADR	Required, Positional 1
ADPTADR	Adapter address	000000000001-7FFFFFFFFFFFFF	Optional

Top

---

### Adapter (ADPTNAME)

Specifies the name of the entry being removed from the adapter file.

The possible values are:

**\*ADPTADR**

The adapter address is used to identify the adapter entry being removed.

*adapter-name*

Specify the name of the adapter entry being removed.

This is a required parameter.

Top

---

### Adapter address (ADPTADR)

Specifies the 12-character hexadecimal adapter address.

Top

---

### Examples

```
RMVLANADPI ADPTNAME(PAYROLL)
```

This command removes the adapter PAYROLL from the adapter file.

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPF8B43**

Request to remove adapter &29 from network adapter file failed. Adapter not found.

#### **CPF8B44**

Adapter address required.

#### **CPF8B62**

Adapter &29 on line &23 not removed. Adapter not found in network adapter file.

#### **CPF8B68**

Line description &23 not found.

#### **CPF8B74**

Request to display active adapters failed.

#### **CPF8B75**

No adapter entries in network adapter file.

#### **CPF8B76**

No functional addresses for adapter.

Top

---

## Remove LAN Adapter (RMVLANADPT)

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

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

The Remove Local Area Network Adapter (RMVLANADPT) command removes an active local area network (LAN) adapter from a line description that is varied on.

### Restrictions:

1. This command is only valid for users with QSECOFR authority.
2. This command is only valid for LAN managers with QSECOFR authority, that are in the controlling mode.

**Note:** The mode (controlling or observing) of the LAN manager is set on the TRNMGRMODE parameter when the line is created or changed with the Create Line Description (Token-Ring Network) (CRTLINTRN) or the Change Line Description (Token-Ring Network) (CHGLINTRN) commands.

**Attention:** Use of this command to remove an active adapter results in loss of communications with any attached products, such as workstations or other systems.

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
LINE	Line description	<i>Name</i>	Required, Positional 1
ADPTNAME	Adapter	<i>Name</i> , *ADPTADR	Required, Positional 2
ADPTADR	Adapter address	<i>Hexadecimal value</i>	Optional

[Top](#)

---

## Line description (LINE)

Specifies the name of the line description attached to the adapter being removed.

This is a required parameter.

[Top](#)

---

## Adapter (ADPTNAME)

Specifies the name of the adapter being removed.

The possible values are:

### \*ADPTADR

The adapter address is used to identify the adapter.

### *adapter-name*

Specify the name of the adapter being removed.

This is a required parameter.

Top

---

## Adapter address (ADPTADR)

Specifies the 12-character hexadecimal adapter address.

Top

---

## Examples

```
RMVLANADPT  LINE(CHGBRANCH) ADPTNAME(*ADPTADR)
             ADPTADR(000000001BFF)
```

This command removes the adapter with an address of 000000001BFF from the line description, CHGBRANCH.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF8B44

Adapter address required.

#### CPF8B47

Request to remove adapter &29 on line &23 failed.

#### CPF8B49

Unable to remove network adapter &29 on line &23.

#### CPF8B50

Adapter &29 on line &23 cannot be removed.

#### CPF8B68

Line description &23 not found.

#### CPF8B70

Network line description &23 not varied on.

#### CPF8B74

Request to display active adapters failed.

#### CPF8B75

No adapter entries in network adapter file.

#### CPF8B76

No functional addresses for adapter.

Top

---

# Remove Library List Entry (RMVLIBLE)

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

Parameters  
Examples  
Error messages

The Remove Library List Entry (RMVLIBLE) command removes a library from the user portion of the library list for the current thread. The user portion is the last portion of the library list. It follows the system portion and, if they exist, any product libraries and the current library entry.

Top

---

## Parameters

Keyword	Description	Choices	Notes
LIB	Library	<i>Name</i>	Required, Positional 1

Top

---

## Library (LIB)

Specifies the library to be removed from the user portion of the library list for the current thread.

This is a required parameter.

*name* Specify the name of the library to be removed from the user portion of the library list for the current thread.

Top

---

## Examples

```
RMVLIBLE LIB(TESTLIB)
```

This command removes the library TESTLIB from the user portion of the library list.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2103

Library &1 already exists in library list.

#### CPF2104

Library &1 not removed from the library list.

#### CPF2106

Library list not available.

**CPF2110**

Library &1 not found.

**CPF2113**

Cannot allocate library &1.

**CPF2118**

Library &1 not added.

**CPF2176**

Library &1 damaged.

**CPF2182**

Not authorized to library &1.

**CPF9807**

One or more libraries in library list deleted.

[Top](#)

---

## Remove License Key Information (RMVLICENSE)

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

Parameters  
Examples  
Error messages

The Remove License Key Information (RMVLICENSE) command can be used to remove software license key information from the license repository for products with keyed compliance. Products with "keyed compliance" require that you have a software license key from the software provider in order to change the usage limit or the expiration date of the license information.

Removing license information from the repository does not affect installed licenses. Any license that is currently being used to access the product on this system remains valid and usable.

**Restrictions:** This command is shipped with public \*EXCLUDE authority.

Top

---

### Parameters

Keyword	Description	Choices	Notes
LICKEYINP	License key input	* <u>LICKEYFILE</u> , *PROMPT	Optional, Positional 1
PRDID	Product identifier	Character value, *ALL	Optional, Positional 2
LICTRM	License term	Character value, * <u>ALL</u>	Optional, Positional 3
FEATURE	Feature	Character value, * <u>ALL</u>	Optional, Positional 4
SERIAL	System serial number	Character value, * <u>LOCAL</u> , *REMOTE, *ALL	Optional
LICKEYFILE	License key file	Qualified object name	Optional
	Qualifier 1: License key file	Name	
	Qualifier 2: Library	Name, * <u>LIBL</u> , *CURLIB	
LICKEYMBR	License key member	Name, * <u>FIRST</u> , *LAST	Optional

Top

---

### License key input (LICKEYINP)

Specifies how the software license key information to be removed is supplied.

#### \*LICKEYFILE

The software license key information is taken from the file specified on the LICKEYFILE parameter.

#### \*PROMPT

The software license key information is supplied through prompting.

Top

---

## Product identifier (PRDID)

Specifies the seven-character identifier of the product for which software license key information is removed.

**\*ALL** The software license key information for all product identifiers is removed.

### *generic\*-product-identifier*

Specify the generic identifier of the products to be removed. A generic product identifier is specified in the same manner as a generic name.

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

### *product-identifier*

Specify the seven-character identifier of the product.

Top

---

## License term (LICTRM)

Specifies the license term for which software license key information is removed.

**\*ALL** The software license key information for all license terms found on the system are removed.

### *license-term*

Specify the license term in Vx, VxRy, or VxRyMz format, where x or y can be a number from 0 through 9, and z can be a number 0 through 9 or a letter A through Z.

Top

---

## Feature (FEATURE)

Specifies the feature of the product specified on the PRDID parameter for which the software license key information is removed.

**\*ALL** The software license key information for all features of the product is removed.

### *feature*

Specify the number of the feature for which software license key information is removed.

Top

---

## System serial number (SERIAL)

Specify the serial number of the system for which software license key information is removed.

### **\*LOCAL**

The software license key information for the local system is removed.

### **\*REMOTE**

The software license key information to be removed is for remote systems only and depends on the value specified for the **License key input (LICKYINP)** parameter.

- If LICKYINP(\*PROMPT) is specified, the software license key information for all remote systems is removed.



- If LICKEYINP(\*LICKKEYFILE) is specified, the software license key information for the remote systems named in the file specified on the LICKKEYFILE parameter is removed.
- \*ALL** The software license key information to be removed is for all systems and depends on the value specified for the LICKEYINP parameter.
- If LICKEYINP(\*PROMPT) is specified, the software license key information for all systems is removed.
  - If LICKEYINP(\*LICKKEYFILE) is specified, the software license key information for all the remote systems named in the file specified on the LICKKEYFILE parameter is removed.

*system-serial-number*

Specify the serial number of the system for which software license key information is removed.

Top

## License key file (LICKKEYFILE)

Specifies the qualified name of the file in which the software license key information to be removed is located. This input file must be in the format of QSYS/QALZAKEY, and can be created by using the LICKKEYFILE parameter on the Display License Key Information (DSPLICKKEY) command.

The name of the license key file can be qualified by one of the following library values:

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

**\*CURLIB**

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

*library-name*

Specify the name of the library to be searched.

The possible value is:

*license-key-file*

Specify the name of the file that contains the software license key information.

Top

## License key member (LICKKEYMBR)

Specifies the name of the member in which the software license key information to be removed is located. This member is in the file specified on the LICKKEYFILE parameter.

**\*FIRST**

The oldest member in the file is used.

**\*LAST**

The newest member in the file is used.

*license-key-member*

Specify the name of the member that contains the software license key information.

Top

## Examples

### Example 1: Removing License Key Information from Prompt Input

```
RMVLICKEY LICKEYINP(*PROMPT) PRDID(1MYPROD)
          LICTRM(V3) FEATURE(5001)
          SERIAL(1234567)
```

This command removes from the license repository the software license key information for the product 1MYPROD, the license term V3, and the feature 5001 for the system with serial number 1234567.

### Example 2: Removing License Key Information from File Input

```
RMVLICKEY LICKEYINP(*LICKEYFILE) SERIAL(*REMOTE)
          LICKEYFILE(*LIBL/MYKEYFILE) LICKEYMBR(*LAST)
```

This command removes from the license repository the software license key information for all the remote systems found in the newest created member of the file MYKEYFILE.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9E55

License Repository object damaged.

#### CPF9E57

&1 license key information records removed, &2 not removed.

#### CPF9E58

License key information not found.

#### CPF9E69

License key information not found in license key file.

Top

---

## Remove Link (RMVLNK)

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

Parameters  
Examples  
Error messages

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

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

- DEL
- ERASE

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

### Restrictions:

1. In the "root" (/), QOpenSys, and user-defined file systems, the user must have write, execute (\*WX) authority to the directory containing the object. If a hard link is to be unlinked, the user must also have object existence (\*OBJEXIST) authority to the object.
2. In the QDLS file system, the user must have all (\*ALL) authority to the object and execute (\*X) authority to the parent directory.
3. The user must have \*X authority to each directory in the path.
4. See the iSeries Security Reference, SC41-5302 book for the authority requirements for other file systems.
5. A user cannot unlink an object within a "root" (/), QOpenSys, or user-defined file system directory that has the "restricted rename and unlink" attribute set on (this attribute is equivalent to the S\_ISVTX mode bit) unless one or more of the following are true:
  - a. The user is the owner of the object.
  - b. The user is the owner of the directory.
  - c. The user has all object (\*ALLOBJ) special authority.
6. A directory cannot be unlinked.
7. The link to a file cannot be removed if the file is a DataLink column in an SQL table and where a row in that SQL table references this file.
8. The restrictions listed above are for the i5/OS objects of the types \*DDIR, \*DSTMF, \*SOCKET, \*STMF, and \*SYMLNK.

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

1. If this command is to be used to remove links for an object that is in these file systems, additional restrictions may apply. To identify these restrictions, see the delete command for the object to be removed. In general, the name of this command is formed using the i5/OS object type value, from the character \* is removed, and add the verb DLT to the beginning. For example, to delete an alert table, which has the object type value of \*ALRTBL, see the Delete Alert Table (DLTALRTBL) command for any additional restrictions.

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

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

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

### QDLS File System Differences

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

Top

---

## Parameters

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

Top

---

## Object link (OBJLNK)

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

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

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Top

---

## Examples

### Example 1: Removing an Object Link

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

This command removes a link named PAY.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFA085

Home directory not found for user &1.

**CPFA093**

Name matching pattern not found.

**CPFA09C**

Not authorized to object. Object is &1.

**CPFA0A1**

An input or output error occurred.

**CPFA0A7**

Path name too long.

**CPFA0A9**

Object not found. Object is &1.

**CPFA0AB**

Operation failed for object. Object is &1.

**CPFA0B1**

Requested operation not allowed. Access problem.

**CPFA0B2**

No objects satisfy request.

**CPFA0BD**

&1 links removed. &2 links failed.

Top



---

## Remove Member (RMVM)

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

Parameters  
Examples  
Error messages

The Remove Member (RMVM) command removes one or more members from the specified physical file or logical file. If the member is removed from a physical file, both the member and the member data are deleted from the system. If the member is removed from a logical file, the member and the access path to the data in the physical file member are deleted from the system.

### Restrictions:

- If a member of another file is sharing the data of the member being deleted, the dependent member must be removed first.
- You must have object existence (\*OBJEXIST) authority for the file that contains the member or members.
- This command is conditionally threadsafe. In multithreaded jobs, this command is not threadsafe for Distributed Data Management (DDM) files of type \*SNA, when SYSTEM(\*RMT) or SYSTEM(\*FILETYPE) is specified.

Top

---

## Parameters

Keyword	Description	Choices	Notes
FILE	Data base file	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Data base file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MBR	Member	<i>Generic name, name, *ALL</i>	Required, Positional 2

Top

---

## Data base file (FILE)

Specifies the physical file or logical file that contains the member or members to be removed.

This is a required parameter.

### Qualifier 1: Data base file

*name* Specify the name of the database file.

### Qualifier 2: Library

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

### \*CURLIB

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

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

---

## Member (MBR)

Specifies the file member or members to be removed.

This is a required parameter.

**\*ALL** All members are to be removed from the specified file.

*generic-name*

Specify the generic name of the members to be removed from the file. A generic name consists of a character string that contains one or more characters followed by an asterisk(\*). If a generic member name is specified, all members in the specified file that have names with the same prefix as the generic member name are removed.

*name* Specify the name of the member to be removed from the file.

Top

---

## Examples

### Example 1: Removing a File Member

```
RMVM FILE(JOBHIST1) MBR(JOBHIST1A)
```

This command removes file member JOBHIST1A from file JOBHIST1. Library list \*LIBL is used to find the file and member. If JOBHIST1 contains other members, they remain unchanged.

### Example 2: Removing Members with Names that Start with SRC

```
RMVM FILE(QGPL/JOBHISTL) MBR(SRC*)
```

This command removes all file members with names that start with SRC from file JOBHISTL in library QGPL.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF32CF

Distributed file error, reason code &3.

#### CPF32C3

Distributed file error, level ID mismatch

#### CPF320A

Member &3 cannot be removed.

#### CPF320B

Operation was not valid for database file &1.

#### CPF3203

Cannot allocate object for file &1 in &2.

#### CPF3220

Cannot do operation on file &1 in &2.



**CPF3273**

File or member not created, deleted or changed.

**CPF7301**

&5 members not removed from file &2 in &3.

**CPF7310**

Member &1 not removed from file &2 in &3.

Top



---

## Remove Mounted FS (RMVMFS)

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

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

The Remove Mounted File System (RMVMFS) command will make a previously mounted file system inaccessible within the integrated file system name space. The file system to be made inaccessible can be a user-defined file system (\*UDFS) on the local system, a remote file system accessed through a Network File System server (\*NFS), or a local or remote NetWare file system (\*NETWARE). If any of the objects in the file system are in use, the command will return an error message to the user. Note that if any part of the file system has itself been mounted over, then this file system cannot be unmounted until it is uncovered.

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

- UNMOUNT

For more information about Network File System commands, see Network File System book, SC41-5714

### Restrictions:

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

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
TYPE	Type of file system	*NFS, *UDFS, *NETWARE, *ALL	Required, Positional 1
MNTOVRDIR	Directory mounted over	<i>Path name</i> , *ALL	Optional
MFS	Mounted file system	<i>Path name</i>	Optional

[Top](#)

---

## Type of file system (TYPE)

Specifies the type of file system to be unmounted.

**\*NFS** The file system to be unmounted is a Network File System. When \*NFS is specified, a directory must be specified for the **Directory mounted over (MNTOVRDIR)** parameter.

### \*UDFS

The file system to be unmounted is a user-defined file system. When \*UDFS is specified, either the MNTOVRDIR or the **Mounted file system (MFS)** parameter may be specified.

### \*NETWARE

The file system to be unmounted is a NetWare file system. When \*NETWARE is specified, a directory must be specified for the MNTOVRDIR parameter.

**\*ALL** File systems of all types are to be unmounted. If \*ALL is specified, a value must be specified for the MNTOVRDIR parameter, and that value may be \*ALL.

This is a required parameter.

Top

---

## Directory mounted over (MNTOVRDIR)

Specifies the path name of the directory that was mounted over ('covered') by a previous ADDMFS (Add Mounted File System) or MOUNT command.

### '*directory-path-name*'

The specified directory that was previously mounted over will be uncovered. If TYPE(\*ALL) was specified, all file systems mounted over the specified directory will be unmounted. If a specific file system type was specified for the **Type of file system (TYPE)** parameter, the file system mounted most recently over the specified directory will be unmounted only if it matches the specified TYPE value.

**\*ALL** All directories that were previously mounted over will be uncovered. If \*ALL is specified, \*ALL must be specified for the TYPE parameter.

This is a required parameter.

Top

---

## Mounted file system (MFS)

Specifies the path name of the file system to be unmounted. This parameter can only be used to unmount a Block Special File (\*BLKSF), when \*UDFS is specified for the **Type of file system (TYPE)** parameter.

Top

---

## Examples

### Example 1: Unmounting a Directory

```
RMVMFS TYPE (*NFS) MNTOVRDIR('/tools')
```

This command unmounts a Network File System that is accessible on directory */tools*.

### Example 2: Unmounting a User-Defined File System

```
RMVMFS TYPE(*UDFS) MFS('/DEV/QASP02/CUST1UDFS')
```

This command unmounts the user-defined file system */dev/qasp02/custudfs*.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFA0A9

Object not found. Object is &1.

#### CPFA1B8

\*IOSYSCFG authority required to use &1.

Top

---

## Remove Message (RMVMSG)

**Where allowed to run:** Compiled CL program or interpreted  
REXX (\*BPGM \*IPGM \*BREXX \*IREXX)  
**Threadsafe:** Yes

Parameters  
Examples  
Error messages

The Remove Message (RMVMSG) command is used by a program to remove the specified message, or a group of messages, from the specified message queue. If an unanswered inquiry message is removed, the default reply is sent before it is removed. If the specified message queue is not allocated to the job in which this command is issued and no other job has the message queue allocated, it is implicitly allocated by this command for the duration of the command.

### Restrictions:

1. To remove a message from the message queue, you must have change (\*CHANGE) authority for the queue and use (\*USE) authority for the library in which the queue is located.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PGMQ	Call stack entry message queue	Single values: *ALLINACT, *EXT Other values: <i>Element list</i>	Optional
	Element 1: Relationship	*SAME, *PRV	
	Element 2: Call stack entry identifier	<i>Element list</i>	
	Element 1: Call stack entry	<i>Character value, _</i>	
	Element 2: Module	<i>Name, *NONE</i>	
	Element 3: Bound program	<i>Name, *NONE</i>	
MSGQ	Message queue	Single values: *PGMQ Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MSGKEY	Message key	<i>Character value</i>	Optional
CLEAR	Clear	*BYKEY, *ALL, *KEEPUNANS, *OLD, *NEW	Optional
RMVEXCP	Remove unhandled exception	*YES, *NO	Optional
RJTDFTRPY	Reject default reply	*NOALWRJT, *ALWRJT	Optional

Top

---

## Call stack entry message queue (PGMQ)

Specifies the call message queue from which the messages are to be removed. Messages can be removed from the external queue (\*EXT) or from a message queue associated with a call stack entry.

### Notes:

1. If CLEAR(\*BYKEY) is specified, the PGMQ parameter is ignored.

2. If values are specified for this parameter, values cannot be specified for the **Message queue (MSGQ)** parameter.

### Single values

#### **\*ALLINACT**

All messages for inactive call stack entries are to be removed from the user's job message queue. If this value is specified, \*ALL must be specified for the **Clear (CLEAR)** parameter.

**\*EXT** The message is to be removed from the external message queue of the job.

### Element 1: Relationship

Two parameter elements are used to specify the call stack entry message queue from which a message is to be removed. The first element specifies whether the message queue is associated with the program or procedure identified by the second element, or if it is associated with the caller of the program or procedure.

#### **\*SAME**

The message is to be removed from the message queue of the program or procedure identified by the second element of this parameter.

**\*PRV** The message is to be removed from the message queue of the program or procedure that called the program or procedure identified by the second element of this parameter.

**Note:** If the message queue previous to the one identified by the second value is for an ILE program entry procedure (PEP), the message will be removed from the message queue immediately previous to the PEP message queue; effectively this would be two message queues previous to the one identified by the program or qualified procedure.

### Element 2: Call stack entry identifier

The second element of this parameter has three elements. Element 1 specifies an OPM program or ILE procedure name or a special value. Element 2 specifies an ILE module name which is used as a qualifier for the value specified in element 1. Element 3 can specify either an OPM program name or an ILE program name or a service program name, depending on what is specified in element 1. Element 3 is also used as a qualifier for what is specified in element 1.

#### Element 1: Call stack entry

**\*** Specifies the OPM program or ILE procedure running this command.

**name** Specify the name of the OPM program or ILE procedure used to identify the call stack entry.

If this element identifies an OPM program, the name specified can be a maximum of 10 characters. If this element identifies an ILE procedure, the name specified can be a maximum of 256 characters.

Nested procedure names can be specified by separating each procedure name with a colon (:). When specifying nested procedure names, the outermost procedure name is identified first, followed by its contained procedures. The innermost procedure name is identified last in the string.

Partial names of programs or procedures can be specified by placing three less-than symbols (<<<) at the beginning of the name or by placing three greater-than symbols (>>>) at the end of the name. If both the greater-than symbols and the less-than symbols are used, the program or procedure name specified is limited to 250 characters.

The system begins its search for the specified program or procedure name with the most recently called program or procedure.

When searching for a partial program or procedure name:

- The less-than symbols (<<<) are truncated when specified only at the beginning of a program or procedure name and the remaining character string is right-justified. The remaining characters in the specified string are compared to the current program or procedure on the call stack, starting with the last position of the program or procedure name and comparing backward.
- The greater-than symbols (>>>) are truncated when specified only at the end of a program or procedure name. The remaining characters in the specified string are compared to the current program or procedure on the call stack, starting with the first position of the program or procedure name.
- The less-than symbols (<<<) and the greater-than symbols (>>>) are truncated when both are specified for a program or procedure name. The remaining characters are used to scan and compare the entire length of the specified string with the current program or procedure on the call stack.

#### Element 2: Module

##### \*NONE

No ILE module qualifier is provided.

*name* Specify the ILE module name to be used to identify the message queue.

#### Element 3: Program

##### \*NONE

No program qualifier is provided.

*name* Specify the program name to be used to identify the message queue.

Top

---

## Message queue (MSGQ)

Specifies the message queue from which one or more messages are to be removed. If this parameter is specified, the **Call stack entry message queue (PGMQ)** parameter cannot be specified.

### Single values

#### \*PGMQ

The call message queue specified for the **Call stack entry message queue (PGMQ)** parameter is the only queue from which the messages are to be removed. If **CLEAR(\*KEEPUNANS)** is specified, **MSGQ(\*PGMQ)** cannot be specified.

### Qualifier 1: Message queue

*name* Specify the name of the message queue from which one or more messages are to be removed.

### Qualifier 2: Library

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

**\*CURLIB**

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

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

Top

---

## Message key (MSGKEY)

Specifies the name of the control language (CL) variable that contains the message reference key of the message to be removed. This parameter can be specified only if CLEAR(\*BYKEY) is specified.

Top

---

## Clear (CLEAR)

Specifies whether one or more messages are to be removed from the queue.

**\*BYKEY**

The message identified by the control language (CL) variable named in the **Message key (MSGKEY)** parameter is to be removed from the message queue.

**\*ALL** All messages are to be removed from the specified message queue.

**\*KEEPUNANS**

All messages except unanswered inquiry messages are to be removed from the specified message queue. If this value is specified, MSGQ(\*PGMQ) cannot be specified, a message queue name must be specified.

**\*OLD** All old messages in the specified message queue are to be removed from the queue.

**\*NEW** All new messages in the specified message queue are to be removed from the queue.

Top

---

## Remove unhandled exception (RMVEXCP)

Specifies the action to be taken when an unhandled exception message is found. An unhandled exception message is an escape, notify, or status message that has been sent to an ILE procedure. When this command is run, the ILE procedure has not yet taken action to tell the system that the exception is handled. One action that the ILE procedure can take is to call a CL program that will remove the exception message. More information on actions that an ILE procedure can take to handle an exception is in the ILE Concepts book, SC41-5606.

This parameter is valid only when working with a message queue that is associated with a call stack entry for an ILE procedure. This parameter is ignored when working with a message queue associated with a call stack entry for an OPM (original program model) program.

**\*YES** The unhandled exception message on the specified message queue is removed. As a result, the exception is handled.

**\*NO** The unhandled exception message on the specified message queue is not removed. The message remains on the queue as an unhandled exception message.

Top



---

## Reject default reply (RJTDFTRPY)

Removing an unanswered inquiry message causes the default reply to be sent to the inquiry message. This value indicates whether a reply handling exit program will be allowed to reject a default reply that is sent as a result of using this command. A reply handling exit program can be registered via the system registration facility for exit point QIBM\_QMH\_REPLY\_INQ.

### \*NOALWRJT

A reply handling exit program will not be allowed to reject a default reply.

### \*ALWRJT

A reply handling exit program will be allowed to reject a default reply. If an exit program rejects the reply, message CPD2476 (Reply rejected by a reply handling exit program) will be sent as a diagnostic message to the program using this command. The CPD2476 will be followed by a CPF2422 (Reply not valid) escape message that the program using this command should monitor for to handle and recover from error situations.

Top

---

## Examples

### Example 1: Removing a Message

```
RMVMSG MSGQ(SMITH) MSGKEY(&KEY)
```

This command removes the message with the reference key specified in the CL variable &KEY from the message queue named SMITH.

### Example 2: Keeping Unanswered Messages

```
RMVMSG MSGQ(SMITH) CLEAR(*KEEPUNANS)
```

This command removes all messages except the unanswered inquiry messages from the message queue named SMITH.

### Example 3: Removing Messages Using a Partial Procedure Name

```
RMVMSG PGMQ(*SAME 'PROCESS_ORDER>>>') CLEAR(*ALL)
```

This command removes all messages from the most recent procedure whose name begins with PROCESS\_ORDER.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF24A6

Value for messages to remove not valid.

#### CPF24AD

Messages to remove must be \*ALL if program message queue is \*ALLINACT.

#### CPF2401

Not authorized to library &1.

#### CPF2403

Message queue &1 in &2 not found.

**CPF2407**  
Message file &1 in &2 not found.

**CPF2408**  
Not authorized to message queue &1.

**CPF241A**  
Clear option &1 in system program is not valid.

**CPF2410**  
Message key not found in message queue &1.

**CPF2411**  
Not authorized to message file &1 in &2.

**CPF2419**  
Message identifier &1 not found in message file &2 in &3.

**CPF2422**  
Reply not valid.

**CPF2450**  
Work station message queue &1 not allocated to job.

**CPF2451**  
Message queue &1 is allocated to another job.

**CPF247A**  
Call stack entry not found.

**CPF2477**  
Message queue &1 currently in use.

**CPF2479**  
Call stack entry not found.

**CPF2483**  
Message file currently in use.

**CPF2499**  
Message identifier &1 not allowed.

**CPF8127**  
&8 damage on message queue &4 in &9. VLIC log-&7.

**CPF8176**  
Message queue for device description &4 damaged.

**CPF9830**  
Cannot assign library &1.

**CPF9838**  
User profile storage limit exceeded.

Top

---

## Remove Message Description (RMVMSGD)

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

Parameters  
Examples  
Error messages

The Remove Message Description (RMVMSGD) command removes a message description from the specified message file.

**Note:** A description of how to print a single message description or a group of message descriptions is in the section entitled *Handling Messages* in the Basic System Operation information in the iSeries Information Center at <http://www.ibm.com/eserver/iseries/infocenter>.

**Restrictions:** You must have use (\*USE) and delete (\*DLT) authorities for the message file.

Top

---

### Parameters

Keyword	Description	Choices	Notes
MSGID	Message identifier	<i>Name</i>	Required, Positional 1
MSGF	Message file	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Message file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Top

---

### Message identifier (MSGID)

Specifies the message identifier of the message to be removed from the message file.

This is a required parameter.

Top

---

### Message file (MSGF)

Specifies the message file containing the message to be removed. Any message file overrides in effect for the job are ignored by this command; the file specified here is the one from which the message is removed.

This is a required parameter.

#### Qualifier 1: Message file

*name* Specify the message file from which the message is to be removed.

#### Qualifier 2: Library

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

### **\*CURLIB**

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

*name* Specify the library where the message-file is located.

Top

---

## **Examples**

```
RMVMSGD MSGID(UIN0115) MSGF(INV)
```

This command removes the message description with the identifier UIN0115 from the message file named INV. The library list is used to find the INV file. Note that if more than one INV message file exists in the libraries being searched, the message description will only be removed from the first INV message file found in the library list.

Top

---

## **Error messages**

### **\*ESCAPE Messages**

#### **CPF2401**

Not authorized to library &1.

#### **CPF2407**

Message file &1 in &2 not found.

#### **CPF2411**

Not authorized to message file &1 in &2.

#### **CPF2419**

Message identifier &1 not found in message file &2 in &3.

#### **CPF2483**

Message file currently in use.

#### **CPF2499**

Message identifier &1 not allowed.

#### **CPF9830**

Cannot assign library &1.

Top

---

## Remove Nickname (RMVNCK)

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

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

The Remove Nickname (RMVNCK) command is used to remove an existing nickname from the system distribution directory. When the removed nickname is no longer available for use, the job issuing this command has ended.

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

### Restrictions:

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

[Top](#)

---

## Parameters

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

[Top](#)

---

## Nickname (NCK)

Specifies the existing nickname that is being removed and the access of the nickname.

The possible nickname value is:

*nickname*

Specify the nickname you are removing.

The possible nickname access values are:

**\*PRIVATE**

The private nickname that you own is being removed.

**\*PUBLIC**

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

This is a required parameter.

[Top](#)

---

## Examples

RMVNCK NCK(SEC44A \*PUBLIC)

This command removes the public nickname SEC44A. If the user has proper authority to the nickname, the nickname is removed.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### **CPF8AA1**

Library QUSRSYS not completely installed.

#### **CPF8360**

Not enough storage for commitment control operation.

#### **CPF9A89**

Nickname function not successful.

#### **CPF905C**

Error occurred trying to find a translation table.

#### **CPF9838**

User profile storage limit exceeded.

[Top](#)

---

## Remove Network Job Entry (RMVNETJOBE)

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

Parameters  
Examples  
Error messages

The Remove Network Job Entry (RMVNETJOBE) command removes a network job entry from the system. The network job entry determines the action taken when a batch input stream is sent to a user on this system using the Submit Network Job (SBMNETJOB) command.

The entry also specifies the user profile that is used for checking the authority to the job description referred to in the batch job. There should be one entry for each user or group of users who submit jobs to this system.

This command is used to remove an entry for a specific user. An entry with a generic ID may still exist that is in effect for that user. For example, if the user removes the entry for user ID, JOE PGMRS, and if there is an entry with a user ID, \*ANY PGMRS or \*ANY \*ANY, that entry is used to handle any jobs submitted by JOE PGMRS. Additional information on the job entry table is in the SNA Distribution Services book, SC41-5410.

### Restrictions:

1. This command is shipped with public \*EXCLUDE authority.
2. The internal value for a node identifier may differ from the characters shown by the RMVNETJOBE command depending on the type of work station (language) being used. If the byte-string value specified for the FROMUSRID command parameter does not match the rules for an internal node identifier value, or if it does not match the internal value for any defined node (ignoring case differences), an error may be reported.

Top

---

## Parameters

Keyword	Description	Choices	Notes
FROMUSRID	User ID	<i>Element list</i>	Required, Positional 1
	Element 1: User ID	<i>Character value</i>	
	Element 2: User ID qualifier	<i>Character value</i>	

Top

---

## User ID (FROMUSRID)

Specifies the two-part user ID that identifies the network job entry that is removed.

This is a required parameter.

Both parts of the user ID are required.

**Note:** Depending on the work station being used, the internal value for a new user identifier may differ from the characters shown by the Display Network Job Entry (DSPNETJOBE) command. If the byte-string

value specified for the FROMUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

[Top](#)

---

## Examples

```
RMVNETJOBE FROMUSRID(JOE SMITH)
```

This command removes the network job entry that is used to determine the action that is taken for any input streams received from user ID, JOE SMITH. The network job authority for user ID, JOE SMITH, is taken from either the network job entry \*ANY SMITH, if that entry exists, or from the network job entry \*ANY \*ANY, if that entry exists. If neither of these entries exist, all jobs received from user ID, JOE SMITH, are rejected.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF8050

Network job table could not be accessed.

#### CPF8051

\*ANY not correct for second part of user ID.

#### CPF8054

Network job entry &1 &2 not removed.

#### CPF9040

Wrong characters used in User ID or address, or List identifier &1 &2.

[Top](#)



---

# Remove Network Table Entry (RMVNETTBLE)

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

Parameters  
Examples  
Error messages

The Remove Network Table Entry (RMVNETTBLE) command is used to remove a network entry from the network table. The network table is used to manage a list of your networks and their associated Internet addresses.

## Restrictions:

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

Top

---

## Parameters

Keyword	Description	Choices	Notes
NETWORK	Network	<i>Character value</i>	Required, Positional 1
INTNETADR	Internet address	<i>Character value</i>	Required, Positional 2

Top

---

## Network (NETWORK)

Specifies the name of the network entry to be removed.

Top

---

## Internet address (INTNETADR)

Specifies the Internet address of the network to be removed. Internet addresses are expressed in the decimal form

`nnn.nnn.nnn.nnn`

where *nnn* is a number ranging from 0 through 255.

Top

---

## Examples

```
RMVNETTBLE NETWORK(NETONE) INTNETADR(9.5.0.0)
```

This command removes the NETONE network entry with address 9.5.0.0 from the network table.

Top

---

## Error messages

### \*ESCAPE Messages

#### TCP1901

Internet address &1 not valid.

#### TCP2649

Interface entry not found.

#### TCP2903

Network entry not found in table.

#### TCP8050

\*IOSYSCFG authority required to use &1.

Top

---

## Remove Node List Entry (RMVNODLE)

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

Parameters  
Examples  
Error messages

The Remove Node List Entry (RMVNODLE) command removes an entry from an existing node list object.

Top

---

### Parameters

Keyword	Description	Choices	Notes
NODL	Node list	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Node list	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *CURLIB, *LIBL</i>	
RMTLOCNAME	Remote location	<i>Element list</i>	Optional
	Element 1: Name or address	<i>Character value</i>	
	Element 2: Type	<i>*SNA, *IP</i>	
CPNAME	SNA node name	Single values: <i>*RMTLOC</i> Other values: <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name, *NETATR</i>	
	Element 2: Control point	<i>Communications name</i>	

Top

---

### Node list (NODL)

Specifies the qualified name of the node list object from which the entry is removed.

The node list name can be qualified by one of the following library values:

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

**\*CURLIB**

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

*library-name*

Specify the name of the library to be searched.

The possible values are:

*node-list-name*

Specify the name of the node list from which the entry is removed.

Top

---

## Remote location (RMTLOCNAME)

Specifies the name and address type of the system to remove from the node list object. The name can be an SNA network ID and control point name, an internet protocol host name, or an internet address.

An SNA node name is specified using the format nnnnnnnn.ccccccc, where nnnnnnnn is the network ID and ccccccc is the control point name. If only the control point name is specified, the local network ID (LCLNETID) network attribute is used as the value of the network identifier (ID) of the system being removed from the node list.

The possible Name or Address value is:

### *remote-location-name*

Specify the remote location name to remove from the node list.

The possible Address Type values are:

**\*SNA** The node name has a Systems Network Architecture (SNA) address type.

**\*IP** The node name has an Internet Protocol (IP) address type.

Top

---

## SNA node name (CPNAME)

Specifies the SNA node name that is being removed from the node list object. This system is specified as two elements: the network ID and the control point name.

### Notes:

1. The RMTLOCNAME parameter is recommended for use in specifying the network ID and the control point name.
2. When the RMTLOCNAME parameter is used to specify the name of a system to remove from the node list, \*RMTLOC must be specified for this parameter.

The possible values are:

### \*RMTLOC

The network ID and control point name are specified using the RMTLOCNAME parameter.

The possible Network ID values are:

### **\*NETATR**

The local network ID (LCLNETID) network attribute is used as the value of the network identifier (ID) of the system being removed from the node list.

### *network-ID*

Specify the network ID of the system to remove from the node list.

The possible Control Point Name value is:

### *control-point-name*

Specify the control point name of the system to remove from the node list.

**Note:** This field is left blank when \*RMTLOC is specified as the network ID.

Top

---

## Examples

### Example 1: Removing a System in the Local Network from a Node List

```
RMVNODLE  NODL(MYLIB/NODL02)  RMTLOCNAME(AS400A01 *SNA)
```

This command removes the entry for system AS400A01, which is in the local network, from the node list NODL02 in library MYLIB. The entry has an SNA address type.

### Example 2: Removing a Host Name from a Node List

```
RMVNODLE  NODL(MYLIB/NODL02)  RMTLOCNAME(MYSYS.NET1.LOCAL *IP)
```

This command removes the entry for host name MYSYS.NET1.LOCAL from the node list NODL02 in library MYLIB. The entry has an address type of IP.

### Example 3: Removing an Internet Address from a Node List

```
RMVNODLE  NODL(MYLIB/NODL02)  RMTLOCNAME('9.13.156.8' *IP)
```

This command removes the entry for internet address 9.13.156.8 from the node list NODL02 in library MYLIB. The entry has an address type of IP.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF7AD4

Network ID &1 not in correct format.

#### CPF7B18

Control point &1 not in correct format.

#### CPF813E

Node list &4 in &9 damaged.

#### CPF96B4

Node list entry does not exist.

#### CPF96B5

Remote location name not in correct format.

#### CPF9801

Object &2 in library &3 not found.

#### CPF9802

Not authorized to object &2 in &3.

#### CPF9803

Cannot allocate object &2 in library &3.

#### CPF9807

One or more libraries in library list deleted.

#### CPF9808

Cannot allocate one or more libraries on library list.

#### CPF9810

Library &1 not found.

#### CPF9820

Not authorized to use library &1.

**CPF9830**

Cannot assign library &1.

[Top](#)

---

## Remove NetWare Aut Entry (RMVNTWAUTE)

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

Parameters  
Examples  
Error messages

The Remove NetWare Authentication Entry (RMVNTWAUTE) command removes an authentication entry from a user profile.

Top

---

### Parameters

Keyword	Description	Choices	Notes
SVRTYPE	Server type	*ALL, *NETWARE3, *NDS	Required, Positional 1
NDSTREE	NDS tree	Character value, *ALL	Optional
SERVER	Server	Character value, *ALL	Optional
USRPRF	User profile	Name, *CURRENT	Optional

Top

---

### Server type (SVRTYPE)

Specifies the type of authentication entry to be removed.

**\*ALL** Remove all authentication entries.

**\*NETWARE3**

Remove entries for NetWare 3.x servers.

**\*NDS** Remove entries for NetWare Directory Services trees.

Top

---

### NDS tree (NDSTREE)

Specifies, for \*NDS entries, the NetWare Directory Services tree name of the authentication entry to be removed.

**\*ALL** Remove authentication entries for all NDS tree names.

*'name'* Specify the name of the NDS tree entry to be deleted.

*'generic-name'*

Specify the generic name of the NDS tree entry to be deleted. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if \*ALL or \*ALLUSR library values can be specified for the name.

---

## Server (SERVER)

Specifies, for \*NETWARE3 entries, the server name of the authentication entry to be removed.

**\*ALL** Remove authentication entries for all server names.

*'name'* Specify the name of the server entry to be deleted.

*'generic-name'*

Specify the generic name of the server entry to be deleted. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if \*ALL or \*ALLUSR library values can be specified for the name.

Top

---

## User profile (USRPRF)

Specifies the user profile from which the authentication entry is to be removed.

**\*CURRENT**

Remove the entry from the current user profile.

*name* Specify the name of the user profile. The user profile must be the current user profile, or the user must have \*USE and \*OBJMGT authority to the user profile, and \*SECADM special authority.

Top

---

## Examples

```
RMVNTWAUTE SVRTYPE(*NDS) NDSTREE(IBMTREE)
```

This command removes the authentication entry for the NetWare Directory Services tree IBMTREE from the current user profile.

Top

---

## Error messages

**\*ESCAPE Messages**

**FPE0219**

Authentication entry not removed for &3.

Top



---

## Remove Server Storage Link (RMVNWSSTGL)

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

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

The Remove Network Server Storage Link (RMVNWSSTGL) command is used to remove an existing client storage space link from a network server description.

### Notes:

1. Removing a client storage space link requires updating items such as network aliases that refer to the drive letter at which the storage space is linked.
2. Removing a client storage space that is linked to a network server description created with **Server connection** specified \*IXSVR or \*ISCSI for the **Network server type (TYPE)** parameter need to take the consideration that if any applications are installed on these drives, the drive may change causing the application to fail.

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
NWSSTG	Network server storage space	<i>Name</i>	Required, Positional 1
NWSD	Network server description	<i>Name</i>	Required, Positional 2
RENUMBER	Renumber link	* <u>NO</u> , *YES	Optional

[Top](#)

---

### Network server storage space (NWSSTG)

Specifies the name of the Network server storage space to be removed from the network server's link list.

This is a required parameter.

[Top](#)

---

### Network server description (NWSD)

Specifies the name of the network server description which contains the link to be removed.

This is a required parameter.

[Top](#)

---

### Renumber link (RENUMBER)

Specifies whether or not the network storage space links are renumbered.

**\*NO** After removing the link of the storage space from the Network server description (NWSD) parameter, the sequence numbers will NOT be renumbered sequentially. Gaps may be present in the sequence.

**\*YES** After removing the link of the storage space from the network server specified (NWSD parameter), the sequence numbers will be renumbered sequentially so that no gaps occur in the sequence.

Top

---

## Examples

### Example 1: Removing a Client Storage Space Link

```
RMVNWSSTGL NWSSTG(PAINTS) NWS(DREMODEL)
```

This command removes the client storage space PAINTS link from the network server REMODEL.

### Example 2: Removing and Renumbering a Client Storage Space Link

```
RMVNWSSTGL NWSSTG(MAIL) NWS(DNTSERVER) RENUMBER(*YES)
```

This command removes the client storage space MAIL link from the network server NTSERVER and renumbers the sequence number of the remaining linked storage spaces to eliminate any gaps in the sequence.

**Note:** For more detail on the affect of renumbering client storage spaces that are linked to a network server description that was created with TYPE(\*IXSVR \*WIN32) or TYPE(\*ISCSI \*WIN32) or TYPE(\*WINDOWSNT) , see the **Windows server on iSeries** topic in the Information Center.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF26B9

Remove network server storage link command failed.

Top

---

## Remove Optical Cartridge (RMVOPTCTG)

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

Parameters  
Examples  
Error messages

The Remove Optical Cartridge (RMVOPTCTG) command can be used to:

- Remove an optical disk cartridge and its volume(s) from an optical device or media library device.
- Remove a cartridge currently in the input/output station of an optical media library.
- Move the input/output station of an optical media library to the out position.
- Open the tray for some optical devices.
- Remove the volume description of an optical volume previously removed using this command with VOLOPT(\*KEEP) specified.

**Restriction:** You must have \*USE authority to use this command. It is shipped with \*EXCLUDE public authority.

Top

---

### Parameters

Keyword	Description	Choices	Notes
VOL	Volume identifier	Character value, *IOSTATION, *MOUNTED, *LEASTUSED, *MAGAZINE	Required, Positional 1
VOLOPT	Volume description option	*REMOVE, *KEEP	Optional, Positional 2
MEDLOC	Media location	*IOSTATION, *MAGAZINE	Optional, Positional 3
RMVCTGLOC	Removed cartridge location	Character value, *NONE	Optional
MLB	Optical media library	Name	Optional
DEV	Optical device	Name	Optional

Top

---

### Volume identifier (VOL)

Specifies the volume identifier of the optical cartridge to be removed from the optical device or media library device. For cartridges with two volumes, only one volume identifier must be specified.

#### \*IOSTATION

The input/output station of an optical media library is moved to the out position. This allows any optical cartridge currently in the input/output station to be removed.

**Note:** This value is only valid when parameter MLB is specified.

#### \*LEASTUSED

The system will determine which optical cartridge is the least recently used and removes it from the specified media library.

**Note:** This value is only valid when parameter MLB is specified.

#### **\*MOUNTED**

The system will remove the description of the volume located in the optical device specified on the DEV parameter. For some optical devices the tray will be opened.

**Note:** This value is only valid when parameter DEV is specified.

#### **\*MAGAZINE**

This value will open the bulk load magazine for removal from the optical library..

**Note:** This value is only valid when parameter MLB is specified.

#### ***volume-identifier***

Specify the volume identifier of the optical cartridge being removed.

Top

---

### **Volume description option (VOLOPT)**

Specifies whether to keep the optical volume description in the optical index database files.

**Note:** This parameter is valid only if a volume identifier is specified on the VOL parameter.

#### **\*REMOVE**

The volume description is removed from the optical index database files.

#### **\*KEEP**

The volume description is kept in the optical index database files. This allows the volume descriptions of removed volumes to be displayed when using the Work with Optical Volumes (WRKOPTVOL) command.

**Note:** This value is only valid for initialized volumes in media library devices. For uninitialized volumes and volumes in other optical devices, this value will be ignored and \*REMOVE will be used.

Top

---

### **Media location (MEDLOC)**

Specifies the final destination of the optical volume being removed.

#### **\*IOSTATION**

The optical cartridge is removed to the input/output station.

#### **\*MAGAZINE**

The optical cartridge is removed to the bulk load magazine.

**Note:** Not all optical libraries are equipped with a bulk load magazine.

Top

---

### **Removed cartridge location (RMVCTGLOC)**

Specifies the external location of the optical cartridge after it is removed.

**Note:** This parameter is valid only when VOLOPT(\*KEEP) is specified.

#### **\*NONE**

No external location is specified.

### *removed-cartridge-location*

Specify the location of the optical cartridge after it is removed. A maximum of 50 characters can be specified.

Top

---

## Optical media library (MLB)

Specifies the name of the optical media library to use for the operation.

**Note:** This parameter is valid only when VOL(\*IOSTATION), VOL(\*MAGAZINE) or VOL(\*LEASTUSED) is specified. This parameter is ignored when a specific volume name is specified on the VOL parameter.

Top

---

## Optical device (DEV)

Specifies the name of the optical device which is to have the volume which it contains removed from the optical index database and tray opened if supported by the device.

**Note:** This parameter is valid only when VOL(\*MOUNTED) is specified. This parameter is ignored when a specific volume name is specified on the VOL parameter.

Top

---

## Examples

### Example 1: Specify a Volume Identifier

```
RMVOPTCTG VOL(VOL01)
```

This command removes optical cartridge with volume identifier VOL01.

### Example 2: Specify a Stand-Alone Device

```
RMVOPTCTG VOL(*MOUNTED) DEV(OPT01)
```

This command removes the volume found in the stand-alone optical device OPT01.

### Example 3: Specify a Media Location of \*MAGAZINE

```
RMVOPTCTG VOL(VOL01) MEDLOC(*MAGAZINE)
```

This command removes optical cartridge with volume identifier VOL01 to the bulk load magazine.

### Example 4: Specify a Volume Name of \*MAGAZINE

```
RMVOPTCTG VOL(*MAGAZINE) MLB(OPTMLB01)
```

This command opens the bulk load magazine so that it can be removed from the optical media library.

Top

---

## Error messages

### \*ESCAPE Messages

- OPT1320**  
Optical volume &1 in use.
- OPT1325**  
Optical volume format not recognized.
- OPT1330**  
Optical volume not found or not useable.
- OPT1331**  
Optical volume &1 not found.
- OPT1346**  
Operation not allowed to volume located in a remote optical device.
- OPT1430**  
Optical volume &1 information removed.
- OPT1460**  
Optical volume &1 is not in an optical device.
- OPT1530**  
&1 does not represent a valid optical device.
- OPT1555**  
Optical device &1 in use.
- OPT1652**  
Device &1 is not an optical media library.
- OPT1671**  
Feature not installed for optical device &1.
- OPT1672**  
Error accessing bulk load magazine for optical device &1.
- OPT1676**  
Bulk load magazine is full for optical device &1.
- OPT1677**  
Bulk load magazine is open for optical device &1.
- OPT1790**  
Operation not allowed or conflicts with another request.
- OPT1805**  
Error accessing optical volume index file.
- OPT1810**  
Error accessing optical directory index file.
- OPT1815**  
Internal program error occurred.
- OPT1820**  
Internal error occurred on optical device &1.
- OPT1821**  
Error occurred on optical device &1.
- OPT1825**  
Optical indexes are incorrect for optical device &1.
- OPT1860**  
Request to optical device &1 failed.

- OPT1861**  
No device description configured for resource &1.
- OPT1862**  
No active device description for resource &1.
- OPT1863**  
Optical libraries need to be reclaimed.
- OPT1872**  
Optical request timed out or was cancelled.
- OPT2301**  
Internal system object in use.
- OPT7740**  
User not authorized to object &2 in library &3 type &4.

[Top](#)





---

# Remove Optical Server (RMVOPTSVR)

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

Parameters  
Examples  
Error messages

The Remove Optical Server (RMVOPTSVR) command disables the ability of hierarchical file system (HFS) APIs to access remote optical servers. The specified servers are no longer accessible to applications using the HFS APIs.

**Restriction:** The user must have \*USE authority to use this command. It is shipped with \*EXCLUDE public authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CSI	Side information	Values (up to 16 repetitions): <i>Name</i> , *ALL	Required, Positional 1
VOLOPT	Volume description option	<u>*REMOVE</u> , *KEEP	Optional, Positional 2

Top

---

## Side information (CSI)

Specifies the communications side information objects representing the remote optical servers to be removed from the optical configuration.

**\*ALL** All remote optical servers in the optical configuration are removed.

***communications-side-information-object-name***

Specify the name of the communications side information object representing the remote optical server. A maximum of 16 names of servers can be specified.

Top

---

## Volume description option (VOLOPT)

Specifies whether to keep the volume descriptions in the optical index database files for optical servers being removed.

**\*REMOVE**

All volume descriptions are removed from the optical index data base files.

**\*KEEP**

All volume descriptions are kept in the optical index database files. This allows \*REMOVED volume descriptions to be displayed when using the Work with Optical Volumes (WRKOPTVOL) command.

Top

---

## Examples

RMVOPTSVR CSI(LAN01)

This command removes optical LAN server LAN01 from the optical configuration. All volume descriptions are removed from the optical index database files.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### **OPT0125**

Command &1 completed with errors, more information in job log.

#### **OPT6730**

No server can be specified when special value \*ALL is used.

[Top](#)

---

## Remove Protocol Table Entry (RMVPCLTBLE)

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

Parameters  
Examples  
Error messages

The Remove Protocol Table Entry (RMVPCLTBLE) command is used to remove a protocol entry from the protocol table. The protocol table is used to manage a list of protocols used in the Internet.

### Restrictions:

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

Top

---

## Parameters

Keyword	Description	Choices	Notes
PROTOCOL	Protocol	<i>Character value</i>	Required, Positional 1

Top

---

## Protocol (PROTOCOL)

Specifies the name of the protocol entry to be removed.

Top

---

## Examples

```
RMVPCLTBLE PROTOCOL(TCP)
```

This command removes the TCP protocol entry from the protocol table.

Top

---

## Error messages

### \*ESCAPE Messages

#### TCP2902

Protocol entry not found in table.

#### TCP8050

\*IOSYSCFG authority required to use &1.

Top



---

## Remove PEX Definition (RMVPEXDFN)

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

Parameters  
Examples  
Error messages

The Remove Performance Explorer Definition (RMVPEXDFN) command removes one or more Performance Explorer definitions from the system. Each definition is stored as a member in the QAPEXDFN file in library QUSRSYS. The member name is the same as the definition name.

### Restrictions:

1. This command is shipped with public \*EXCLUDE authority.
2. To use this command you must have \*SERVICE special authority, or be authorized to the Service Trace function of i5/OS through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM\_SERVICE\_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
3. The following user profiles have private authorities to use the command:
  - QPGMR
  - QSRV
4. Two threads within the same job will not be allowed to run RMVPEXDFN at the same time. The thread that issued RMVPEXDFN first will run the command to completion while the second RMVPEXDFN waits.

Top

---

## Parameters

Keyword	Description	Choices	Notes
DFN	Definition	Generic name, name, *ALL	Required, Positional 1

Top

---

## Definition (DFN)

Specifies the name of the Performance Explorer definition being removed. A specific or generic definition name, or \*ALL, can be specified.

**\*ALL** All Performance Explorer definitions are removed.

### *definition-name*

Specify the name of the Performance Explorer definition to be removed.

### *generic\*-definition-name*

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

Top

---

## Examples

### Example 1: Removing a Single Definition

```
RMVPEXDFN DFN(SAMPLE)
```

This command removes the member named SAMPLE from file QAPEXDFN in library QUSRSYS that contains the performance explorer definition named SAMPLE.

### Example 2: Removing All Definitions that Start with SAM

```
RMVPEXDFN DFN(SAM*)
```

This command removes all definitions with names that start with SAM by removing all members that start with SAM from file QAPEXDFN in library QUSRSYS.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF3203

Cannot allocate object for file &1 in &2.

#### CPF3220

Cannot do operation on file &1 in &2.

#### CPF7301

&5 members not removed from file &2 in &3.

#### CPF7310

Member &1 not removed from file &2 in &3.

[Top](#)

---

## Remove PEX Filter (RMVPEXFTR)

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

Parameters  
Examples  
Error messages

The Remove Performance Explorer Filter (RMVPEXFTR) command removes one or more Performance Explorer filters from the system. Each filter is stored as a member in the QAPEXFTR file in library QUSRSYS. The member name is the same as the filter name.

### Restrictions:

1. This command is shipped with public \*EXCLUDE authority.
2. To use this command you must have \*SERVICE special authority, or be authorized to the Service Trace function of i5/OS through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM\_SERVICE\_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
3. The following user profiles have private authorities to use the command:
  - QPGMR
  - QSRV
4. Two threads within the same job will not be allowed to run RMVPEXFTR at the same time. The thread that issued RMVPEXFTR first will run the command to completion while the second RMVPEXFTR waits.

Top

---

## Parameters

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

Top

---

## Filter (FTR)

Specifies the name of the Performance Explorer filter being removed. A specific or generic filter name, or \*ALL, can be specified.

**\*ALL** All Performance Explorer filters are removed.

### *generic\*-filter-name*

Specify the generic name of the Performance Explorer filter to be removed. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. For more information about generic object names, see the i5/OS objects topic under Programming->CL in the iSeries Information Center at the Web site: <http://www.ibm.com/eserver/iseries/infocenter>

### *filter-name*

Specify the name of the Performance Explorer filter to be removed.

---

## Examples

### Example 1: Removing a Single Filter

```
RMVPEXFTR  FTR(SAMPLE)
```

This command removes the member named SAMPLE from file QAPEXFTR in library QUSRSYS that contains the performance explorer filter named SAMPLE.

### Example 2: Removing All Filters that Start with SAM

```
RMVPEXFTR  FTR(SAM*)
```

This command removes all filters with names that start with SAM by removing all members that start with SAM from file QAPEXFTR in library QUSRSYS.

---

## Error messages

### \*ESCAPE Messages

#### CPF3203

Cannot allocate object for file &1 in &2.

#### CPF3220

Cannot do operation on file &1 in &2.

#### CPF7301

&5 members not removed from file &2 in &3.

#### CPF7310

Member &1 not removed from file &2 in &3.



---

## Remove PF Constraint (RMVPFCST)

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

Parameters  
Examples  
Error messages

The Remove Physical File Constraint (RMVPFCST) command can be used to remove one or more constraint relationships between physical files. The constraint relationships that can be removed with this command are referential constraints, unique constraints, primary key constraints, and check constraints.

### Restrictions:

- You must have object management (\*OBJMGT) or object alter (\*OBJALTER) authority to the physical file specified for the **File (FILE)** parameter.
- You must have execute (\*EXECUTE) authority to the library that contains the physical file.
- You cannot remove a constraint relationship from a file that your user job has open.
- This command is conditionally threadsafe. In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type \*SNA.

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>	
CST	Constraint name	Single values: *ALL, *CHKPND Other values (up to 300 repetitions): <i>Character value</i>	Required, Positional 2
TYPE	Constraint type	<b>*ALL</b> , *REFCST, *UNQCST, *PRIKEY, *CHKCST	Optional, Positional 3
RMVCST	Remove constraint	<b>*RESTRICT</b> , *REMOVE, *KEEP	Optional

Top

---

## File (FILE)

Specifies the physical file from which a constraint is to be removed. For a referential constraint, this file can be a dependent file only.

This is a required parameter.

### Qualifier 1: File

*name* Specify the name of the 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 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

---

## Constraint name (CST)

Specifies the name of the constraint relationship being removed.

This is a required parameter.

### Single values

\*ALL All of the constraint relationships for the file specified on the FILE parameter are removed.

### \*CHKPND

The constraint relationships that have records that are possibly in violation of the constraints (check pending) are removed. A **check pending** state occurs when the system has not yet determined that the values of a dependent file are all valid in relation to its parent key or record values for the check constraint expression are valid. Only referential and check constraints can be in check pending.

### Other values (up to 300 repetitions)

#### *character-value*

Specify the name of the constraint.

**Note:** The case is preserved when lowercase characters are specified.

Top

---

## Constraint type (TYPE)

Specifies the type of constraint relationship named on the CST parameter that is being removed from the physical file.

\*ALL All types of primary key and unique constraints are removed. The following are examples of the CST parameter dependencies:

- CST(\*ALL) TYPE(\*ALL) - All constraints are removed
- CST(\*CHKPND) TYPE(\*ALL) - All referential and check constraints in check pending are removed
- CST(ABC) TYPE(\*ALL) - The constraint ABC is removed

**Note:** This value is ignored for referential constraints when the RMVCST parameter is specified.

### \*REFCST

The referential constraints are removed. The following are examples of the CST parameter dependencies:

- CST(\*ALL) TYPE(\*REFCST) - All referential constraints are removed
- CST(\*CHKPND) TYPE(\*REFCST) - All referential constraints in check pending are removed
- CST(ABC) TYPE(\*REFCST) - The referential constraint ABC is removed

### \*UNQCST

The unique constraints are removed.

**Note:** If the unique constraint is a primary key, the unique constraint is removed, but the primary key definition and the file's access path are not removed.

The following are examples of the CST parameter dependencies:

- CST(\*ALL) TYPE(\*UNQCST) - All unique constraints (except the primary key constraint) are removed
- CST(\*CHKPND) TYPE(\*UNQCST) - Not valid; unique constraints cannot be in check pending
- CST(ABC) TYPE(\*UNQCST) - The unique constraint ABC is removed (unless it is a primary key constraint)

#### **\*PRIKEY**

The primary key constraint is removed. The following are examples of the CST parameter dependencies:

- CST(\*ALL) TYPE(\*PRIKEY) - The primary key constraint is removed
- CST(\*CHKPND) TYPE(\*PRIKEY) - Not valid; primary key constraints cannot be in check pending
- CST(ABC) TYPE(\*PRIKEY) - The primary key constraint ABC is removed

#### **\*CHKCST**

The check constraints are removed. The following are examples of the CST parameter dependencies:

- CST(\*ALL) TYPE(\*CHKCST) - All check constraints are removed
- CST(\*CHKPND) TYPE(\*CHKCST) - All check constraints in check pending are removed
- CST(XYZ) TYPE(\*CHKCST) - The check constraint XYZ is removed

Top

---

## **Remove constraint (RMVCST)**

Specifies how much of the constraint relationship on the dependent file is removed when a primary key or unique constraint that is also a parent key is removed from the parent file of a referential constraint.

**Note:** This parameter is ignored if \*REFCST is specified for the **Constraint type (TYPE)** parameter.

#### **\*RESTRICT**

The constraint is not removed if the constraint is either defined or established between the parent file and the dependent file. Neither the foreign key access path nor the foreign key of the dependent file is removed.

#### **\*REMOVE**

The constraint and the constraint definition between the parent file and the dependent file are removed. The corresponding foreign key is removed. The foreign key access path of the dependent file is removed only if one exists and is not shared.

#### **\*KEEP**

The constraint between the parent file and the dependent file is removed, but the constraint definition is not removed. The corresponding foreign key and the foreign key access path of the dependent file are not removed.

Top

---

## **Examples**

In these examples, the unique constraint UNIQUE\_Department\_NUMBER and the referential constraint EMPLOYEE\_Department were added to the files by issuing the following Add Physical File Constraint (ADDPFCST) commands:

```
ADDPFCST  FILE(MYLIB/DEPARTMENTS)  TYPE(*UNQCST)
          KEY(DEPTNUM)  CST(UNIQUE_Department_NUMBER)
ADDPFCST  FILE(MYLIB/PERSONNEL)  TYPE(*REFCST)  KEY(DEPTNO)
          CST(EMPLOYEE_Department)
```

### Example 1: Removing a Unique Constraint

```
RMVPCST  FILE(MYLIB/DEPARTMENTS)  CST(*ALL)  TYPE(*ALL)
```

This command removes the unique constraint UNIQUE\_Department\_NUMBER from the file DEPARTMENTS located in the library MYLIB.

### Example 2: Removing a Referential Constraint

```
RMVPCST  FILE(MYLIB/PERSONNEL)  CST(EMPLOYEE_Department)
          TYPE(*REFCST)  RMVPCST(*RESTRICT)
```

This command removes the referential constraint EMPLOYEE\_Department from the dependent file PERSONNEL located in the library MYLIB. Because the parent file had not yet been established (the PRNFILE had not been specified on the ADDPFCST command) the removal is not restricted.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF32B1

Constraint(s) not removed from file &1.

Top

---

## Remove Physical File Trigger (RMVPFTRG)

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

Parameters  
Examples  
Error messages

The Remove Physical File Trigger (RMVPFTRG) command removes the triggers that call trigger programs from a specified physical file. The triggers to be removed can be specified by trigger events, trigger times or trigger name. A trigger program is a program that has been added to the specified physical file by the Add Physical File Trigger (ADDPFTRG) command (system trigger) or the SQL CREATE TRIGGER statement (SQL trigger).

If a trigger time, trigger event and trigger name are specified and the trigger time and event do not match the time and event for the trigger definition name, the trigger will not be removed.

Once a trigger is removed from the physical file for a specified trigger time or event or name, the trigger program is no longer called when the trigger event occurs within the file. If the trigger was a system trigger, the trigger program continues to exist on the system. If the trigger was an SQL trigger, the trigger program is deleted.

An exclusive-no-read lock is held on the physical file when removing the trigger from that file. All logical files which are built over the physical file are also held with the exclusive no-read lock.

### Restrictions:

- You must have object alter (\*OBJALTER) or object management (\*OBJMGT) authority to the physical file and execute (\*EXECUTE) authority to the library that contains the file.
- If the physical file or a dependent logical file or Structured Query Language (SQL) view is opened in this or another job, the trigger cannot be removed.
- While this command is running, neither the physical file nor any dependent logical files can be opened.
- This command is conditionally threadsafe. In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type \*SNA.

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>	
TRGTIME	Trigger time	<i>*ALL, *BEFORE, *AFTER</i>	Optional, Positional 2
TRGEVENT	Trigger event	<i>*ALL, *INSERT, *DELETE, *UPDATE, *READ</i>	Optional, Positional 3
TRG	Trigger	<i>Character value, *ALL</i>	Optional
TRGLIB	Trigger library	<i>Name, *FILE, *CURLIB</i>	Optional

Top

---

## Physical file (FILE)

Specifies the physical file from which the trigger is to be removed. The specified file must exist on the system.

This is a required parameter.

### Qualifier 1: Physical file

*name* Specify the name of the file from which the trigger is to be removed.

### 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

---

## Trigger time (TRGTIME)

Specifies the trigger to be removed, based on the time when the trigger program is called.

**\*ALL** All triggers for programs called either before or after a trigger event are removed.

#### **\*BEFORE**

The triggers for programs called before a trigger event are removed.

#### **\*AFTER**

The triggers for programs called after a trigger event are removed.

Top

---

## Trigger event (TRGEVENT)

Specifies the trigger event for which the associated triggers are removed.

**\*ALL** All triggers for insert, delete, and update operations are removed.

#### **\*INSERT**

The triggers for insert operations are removed.

#### **\*DELETE**

The triggers for delete operations are removed.

#### **\*UPDATE**

The triggers for update operations are removed.

#### **\*READ**

The triggers for read operations are removed.

Top

---

## Trigger (TRG)

Specifies the name of the trigger being removed.

**\*ALL** All trigger definitions are removed, including SQL triggers.

**character-value**

Specify the name of the trigger. You can specify a maximum of 128 characters without delimiters, or 258 characters with quotation mark (") delimiters.

The following are examples of the TRG parameter dependencies:

- TRG(\*ALL) TRGTIME(\*ALL) TRGEVENT(\*ALL) - All triggers are removed.
- TRG(\*ALL) TRGTIME(\*INSERT) TRGEVENT(\*AFTER) - All \*INSERT \*AFTER triggers are removed.
- TRG(ABC) TRGTIME(\*ALL) TRGEVENT(\*ALL) - The triggers named ABC are removed.
- TRG(ABC) TRGTIME(\*BEFORE) TRGEVENT(\*INSERT) - The trigger named ABC is removed if the trigger time is \*BEFORE and the triggered event is \*INSERT. If the trigger time and triggered event do not match, trigger ABC will not be removed.

Top

---

## Trigger library (TRGLIB)

Specifies the library for the trigger being removed.

**Note:** The special values \*LIBL and \*CURLIB are the values of the job running when the trigger is removed.

**\*FILE** The library for the file specified for the **Physical file (FILE)** parameter is used.

**\*CURLIB**

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

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

Top

---

## Examples

### Example 1: Removing All Triggers for Insert Events

```
RMVPFTRG FILE(EMP) TRGEVENT(*INSERT)
```

This command removes all triggers for programs called by insert operations from the physical file named EMP.

### Example 2: Removing All Triggers for Programs Called Before a Trigger Event

```
RMVPFTRG FILE(EMP) TRGTIME(*BEFORE)
```

This command removes all triggers for programs called before trigger events from the physical file named EMP.

### Example 3: Removing a Trigger for a Program Called After an Insert Event

```
RMVPFTRG FILE(EMP) TRGTIME(*AFTER) TRGEVENT(*INSERT)
```

This command removes the trigger for the program called after an insert operation from the physical file named EMP.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF32C6

Trigger operation not successful.

[Top](#)



---

## Remove Program (RMVPGM)

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

Parameters  
Examples  
Error messages

The Remove Program (RMVPGM) command removes one or more programs from the current debugging session. All breakpoints and traces defined in each program are removed, and the programs are returned to their normal state. If a program is added again, breakpoints and traces must be specified again.

### Restrictions:

1. You can use this command only in debug mode. To start debug mode, refer to the Start Debug (STRDBG) command.
2. You cannot use this command to remove bound programs from a debugging session.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PGM	Program	Single values: *DFTPGM, *ALL Other values (up to 20 repetitions): <i>Name</i>	Optional, Positional 1

Top

---

## Program (PGM)

Specifies which programs are removed from the current debugging session.

### Single values

#### \*DFTPGM

The program currently specified as the default program in the debugging session is the program to be removed. The debugging session no longer has a default program unless one is specified later.

\*ALL All the programs that are currently in debug mode are removed.

### Other values (up to 20 repetitions)

*name* Specify the name of the program to be removed from the current debugging session.

Top

---

## Examples

```
RMVPGM PGM(PGMX PGMY PGMZ)
```

This command removes the three programs PGMX, PGMY, and PGMZ from the current debugging session. All breakpoints and data traces are removed from the programs.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1999

Errors occurred on command.

[Top](#)

---

## Remove Prestart Job Entry (RMVPJE)

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

Parameters  
Examples  
Error messages

The Remove Prestart Job Entry (RMVPJE) command removes a prestart job entry from the specified subsystem description.

When removing an entry where \*LIBL is specified for the library name, the library list is searched for a program with the specified name. If a program is found in the library list but an entry exists with a different library name (which is found later in the library list), no entry is removed. If a program is not found in the library list but an entry exists, no entry is removed.

### Restrictions:

- To use this command, you must have:
  - object operational (\*OBJOPR), object management (\*OBJMGT), and read (\*READ) authority to the specified subsystem description and execute (\*EXECUTE) authority to the library containing the subsystem description.
- If the prestart job is active, the End Prestart Jobs (ENDPJ) command must be run before this command can be run.

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>	
PGM	Program	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Program	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Top

---

## Subsystem description (SBSD)

Specifies the name and library of the subsystem description that contains the prestart job entry that is removed.

This is a required parameter.

### Qualifier 1: Subsystem description

*name* Specify the name of the subsystem description from which the prestart job entry is being removed.

### Qualifier 2: Library

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

**\*CURLIB**

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

*name* Specify the subsystem description's library from which the prestart job entry is being removed.

Top

---

## Program (PGM)

Specifies the name and library of the program for the prestart job entry that is removed. Two entries with the same program name can exist in a single subsystem description but they must have different library names.

This is a required parameter.

**Qualifier 1: Program**

*name* Specify the name of the program run by the prestart job.

**Qualifier 2: Library**

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

**\*CURLIB**

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

*name* Specify the library of the program run by the prestart job.

Top

---

## Examples

```
RMVPJE  SBSDB(QGPL/PJE)  PGM(QGPL/PGM1)
```

This command removes the prestart job entry for the PGM1 program (in the QGPL library) from the PJE subsystem description contained in the QGPL library.

Top

---

## Error messages

**\*ESCAPE Messages**

**CPF1619**

Subsystem description &1 in library &2 damaged.

**CPF1691**

Active subsystem description may or may not have changed.

**CPF1697**

Subsystem description &1 not changed.

Top

---

# Remove Program Temporary Fix (RMVPTF)

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

Parameters  
Examples  
Error messages

The Remove Program Temporary Fix (RMVPTF) command removes the specified program temporary fixes (PTFs) from the specified product. If the PTFs are temporarily applied, the original objects that they replaced are returned. The PTFs can be temporarily removed, in which case they are held in the product PTF library and they can be applied later. If the PTFs have not been applied, they can be permanently removed and moved to the QRPLOBJ library.

The RMVPTF command is used to remove immediate PTFs at the time the command is run, or to request PTFs to be removed during the next unattended initial program load (IPL).

**Restriction:** To use this command, you must be signed on as QSRV, or have \*ALLOBJ authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
LICPGM	Product	<i>Character value</i>	Required, Positional 1
RLS	Release	<i>Character value, *ONLY</i>	Optional
SELECT	PTF numbers to select	Single values: <b>*ALL</b> Other values (up to 300 repetitions): <i>Character value</i>	Optional
OMIT	PTF numbers to omit	Values (up to 300 repetitions): <i>Character value</i>	Optional
RMV	Extent of change	<b>*TEMP, *PERM</b>	Optional
DELAYED	Delayed PTFs	<b>*NO, *YES</b>	Optional
IPLRMV	Remove on unattended IPL	<b>*YES, *NO</b>	Optional
RMVDEP	Remove dependent PTFs	<b>*NO, *YES</b>	Optional

Top

---

## Product (LICPGM)

Specifies the 7-character identifier of the product from which the PTFs are removed.

This is a required parameter.

Top

---

## Release (RLS)

Specifies the release level of the PTFs being loaded.

The possible values are:

### \*ONLY

This value is valid when only one release of the product's base option is installed on the system. PTFs for all installed options of the product are removed regardless of the release-level of the option.

### *release-level*

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

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

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

Top

---

## PTF numbers to select (SELECT)

Specifies which PTFs are removed from the specified product. The **PTF numbers to omit (OMIT)** parameter cannot be specified if single PTF numbers are specified in the **PTF numbers to select (SELECT)** parameter.

You can enter multiple values for this parameter.

The possible values are:

\*ALL All the PTFs are removed from the product. Those that were permanently applied are ignored by this command. If all PTFs cannot be removed, messages are sent to the operator indicating the PTFs that are not removed and the reasons why they are not being removed.

### *PTF-number*

Specify the PTF identification number of each program temporary fix being removed. A maximum of 300 PTF numbers can be specified.

Top

---

## PTF numbers to omit (OMIT)

Specifies that all PTFs are removed except for those specified in this parameter. Specify the PTF numbers of the program temporary fixes that are omitted (left in the system) when all the rest are removed. A maximum of 300 PTF numbers can be specified. The OMIT parameter cannot be specified if single PTF numbers are specified in the **PTF numbers to select (SELECT)** parameter.

You can enter multiple values for this parameter.

Top

---

## Extent of change (RMV)

Specifies whether the PTFs are removed temporarily or permanently. Permanently removed PTF objects are moved to QRPLOBJ or deleted. Temporarily removed PTF objects are held in the product PTF library for application at a later time.

The possible values are:

### \*TEMP

The PTFs are removed and held in the product PTF library so that they can be applied again later.

### \*PERM

The PTFs are permanently removed and placed in QRPLOBJ.

Top

---

## Delayed PTFs (DELAYED)

Specifies whether immediate PTFs are removed at the time the command is run, or whether immediate and delayed PTFs are removed in a delayed manner during the next unattended IPL.

\*NO Any immediate PTFs identified are removed at the time the command is processed. For delayed PTFs, if the PTF has a status of **Not applied** and you have specified RMV(\*PERM), the PTF is permanently removed at the time the command is run. All other delayed PTFs are ignored during the RMVPTF request and are not removed. For immediate PTFs that are being temporarily removed, any preconditions that are active will result in the PTF not being removed. A message is sent for each PTF that is not removed.

\*YES Both delayed and immediate PTFs are removed during the next unattended IPL. The **Remove on unattended IPL (IPLRMV)** parameter determines whether the PTFs are removed during the next unattended IPL, or whether any previous request to remove the PTFs during the next unattended IPL is canceled.

Top

---

## Remove on unattended IPL (IPLRMV)

Specifies the action that is done for delayed or immediate PTFs at the next unattended IPL. This parameter is valid only if \*YES was specified for the **Delayed PTFs (DELAYED)** parameter.

The possible values are:

\*YES The identified PTFs are removed at the next unattended IPL. The **Extent of change (RMV)** parameter determines whether the removal is temporary or permanent.

\*NO Any previous request to remove the identified PTFs at the next unattended IPL is canceled.

Top

---

## Remove dependent PTFs (RMVDEP)

Specifies whether dependent PTFs and mutually dependent PTFs in the same product and option as the PTFs specified in the SELECT parameter are processed with the PTFs specified in the SELECT parameter list. The RMVDEP parameter is valid only if \*ALL is not specified in the SELECT parameter.

The possible values are:

\*NO The dependent and mutually dependent PTFs are not processed with the SELECT parameter list. No PTFs are removed if any PTF specified in the list has dependent PTFs not also in the list or already applied. Messages identify the missing dependent PTFs and the specified PTFs on which they depend.

\*YES The dependent and mutually dependent PTFs are removed with the SELECT parameter list.

Top

---

## Examples

### Example 1: Temporarily Removing PTFs

```
RMVPTF LICPGM(5722SS1) DELAYED(*YES)
```

This command temporarily removes all temporarily applied PTFs from the operating system (5722SS1) at the next IPL. The PTFs can be applied again, if necessary, using the APYPTF command.

### Example 2: Permanently Removing PTFs

```
RMVPTF LICPGM(5722SS1) SELECT(SI10002 SI10005) RMV(*PERM)
```

This command permanently removes two PTFs (numbers SI10002 and SI10005) from the operating system (5722SS1). The two PTFs are moved to QRPLOBJ and must be loaded again using the LODPTF command before they can be applied.

### Example 3: Removing PTFs and Their Mutual Dependents

```
RMVPTF LICPGM(5722SS1) SELECT(SI00003 SI00008 SI00012)  
      DELAYED(*YES) RMVDEP(*YES)
```

This command temporarily removes PTFs SI00003, SI00008, SI00012, and their dependent and mutually dependent PTFs within the same product and option from the operating system in library QSYS at the next IPL.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF24B4

Severe error while addressing parameter list.

#### CPF35A0

Cannot allocate library &1.

#### CPF35A1

Wrong copy of Licensed Internal Code in use.

#### CPF35A4

Licensed Internal Code fix &2 cannot be removed.

#### CPF35A9

Error occurred while processing Licensed Internal Code fix.

#### CPF35C0

IPL action cannot be removed for PTF &1-&2 &3.

#### CPF35D0

Licensed Internal Code fix &1-&2 &3 not set to be removed permanently.

#### CPF35D2

PTF &1-&2 not removed.

#### CPF35EB

Multiple releases of product &1 installed.

#### CPF35E4

Information for PTF &1-&2 &3 not complete.



**CPF35FB**  
PTF &1-&2 not removed.

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

**CPF3564**  
PTF &1-&2 damaged.

**CPF358A**  
Release not valid.

**CPF3596**  
PTF numbers in select/omit list not permitted.

**CPF3598**  
PTF function already in process.

**CPF3602**  
PTF &2 not removed because it is permanently applied.

**CPF3604**  
PTF not removed because an error occurred.

**CPF3606**  
Product &1 &2 not installed.

**CPF361E**  
Error occurred while removing PTFs for product &1.

**CPF3612**  
Library &1 not found.

**CPF3641**  
No immediate PTFs removed.

**CPF3658**  
No program temporary fixes identified.

**CPF3693**  
Service function ended because error occurred.

**CPF3931**  
Required programs not found. PTF incomplete.

**CPF3945**  
Records of PTF activity for licensed program are deleted.

**CPF3956**  
Error occurred during PTF processing.

Top



---

## Remove RDB Directory Entry (RMVRDBDIRE)

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

Parameters  
Examples  
Error messages

The Remove Relational Database Directory Entry (RMVRDBDIRE) command removes a specific entry, generic entries, all entries, or all remote entries from the relational database (RDB) directory.

### Restrictions:

\*IOSYSCFG special authority is needed to remove the \*LOCAL RDB entry.

Top

---

## Parameters

Keyword	Description	Choices	Notes
RDB	Entry	Generic name, name, *ALL, *ALLRMT	Required, Positional 1

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:

### \*ALLRMT

All entries except the \*LOCAL entry in the RDB directory are removed.

\*ALL All entries in the RDB directory are removed.

### *generic\*-relational-database-name*

Specify the generic name of the RDB entries to be removed. 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 RDB entry to be removed.

Top

---

## Examples

```
RMVRDBDIRE RDB(YOURRDB)
```

This command removes the entry YOURRDB from the relational database directory. The entry is no longer accessible.

---

## Error messages

### \*ESCAPE Messages

#### CPF3EC2

Remove relational database directory entry failed.

---

## Remove REXX Buffer (RMVREXBUF)

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

Parameters  
Examples  
Error messages

The Remove REXX Buffer (RMVREXBUF) command removes one or more buffers from the REXX external data queue.

Top

---

### Parameters

Keyword	Description	Choices	Notes
BUFFER	Buffer	Decimal number, <u>*CURRENT</u> , *ALL	Optional, Positional 1

Top

---

### Buffer (BUFFER)

Specifies the number of the buffer to remove. The buffer identified by the number and all buffers added after it, are removed.

The possible values are:

**\*CURRENT**

Only the current buffer is removed.

**\*ALL** All buffers and entries are removed from the REXX external data queue. This is equivalent to specifying BUFFER(0).

***variable-name***

Specify a variable name. In a control language (CL) program, a decimal variable with a minimum length of 11 digits and no decimal positions must be specified.

***buffer-number***

Specify the number of the buffer to remove.

Top

---

### Examples

```
RMVREXBUF  BUFFER(2)
```

This command removes buffer number 2 and all buffers with numbers higher than 2 from the REXX external data queue.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF7CF7

REXX external data queue is damaged.

[Top](#)

---

## Remove Remote Definition (RMVRMTDFN)

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

Parameters  
Examples  
Error messages

The Remove Remote Definition (RMVRMTDFN) command is used to remove the definition of attributes for a remote system.

**Restriction:** The user must have \*ALLOBJ authority.

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>	

Top

---

### System name (SYSTEM)

Specifies the system name and system group of the remote system being removed.

The possible values are:

**\*ANY** Removes the default definition for a remote system not covered by the other entries.

**\*ALL** Removes the definitions for all remote systems.

The possible **System Name** value is:

*system-name*

Specify the name of the remote system being removed.

The possible **System Group** value is:

*system-group*

Specify the group name of the remote system being removed. The group name is blank if this value is not specified.

Top

---

### Examples

#### Example 1: Removing a Specific Remote Definition

```
RMVRMTDFN SYSTEM(RCHAS1)
```

This command removes the definition from remote system RCHAS1. This system now uses the values for the \*ANY remote definition or the defaults.

## Example 2: Removing all Remote Definitions

```
RMVRMTDFN  SYSTEM(*ALL)
```

This command removes all remote system definitions. All systems now use the default values.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF6DCA

SYSTEM parameter cannot be local system.

#### CPF6DCC

Remote definition for system &1 &2 not found.

#### CPF9899

Error occurred during processing of command.

[Top](#)



---

## Remove Remote Journal (RMVRMTJRN)

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

Parameters  
Examples  
Error messages

The Remove Remote Journal (RMVRMTJRN) command disassociates a remote journal on the specified target system from the specified journal on the source system. The journal on the source system may be either a local journal or another remote journal.

The remote journal, and any associated journal receivers, are not deleted from the target system by the command processing. No processing is performed on the target system for the command. The remote journal that remains on the target system may later be added back to the remote journal definition for the journal on the source system by using the Add Remote Journal (ADDRMTJRN) command or the Add Remote Journal (QjoAddRemoteJournal) API.

It is the responsibility of the user to delete the remote journal and any associated journal receivers from the target system, if so desired.

Once a remote journal association has been removed from a journal, all of the journal receivers that are currently in the journal's receiver directory on the source system will no longer be protected from deletion even if the journal entries have not yet been replicated to the remote journal.

### Restrictions:

- The command must be called from the source system for a local or remote journal.
- The remote journal on the specified target system cannot have a journal state of \*ACTIVE.
- The specified relational database (RDB) directory entry must meet the following rules:
  - The communications protocol must be one of the remote journal function supported protocols.
  - The remote location name in the RDB cannot refer to the \*LOCAL database.
  - The RDB cannot use an application requester driver program (\*ARDPGM) to locate the target system.

Top

---

## Parameters

Keyword	Description	Choices	Notes
RDB	Relational database	<i>Name</i>	Required, Positional 1
SRCJRN	Source journal	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Source journal	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
TGTJRN	Target journal	Single values: <b>*SRCJRN</b> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Target journal	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i>	

Top

---

## Relational database (RDB)

The name of the relational database directory entry that contains the remote location name of the target system.

This is a required parameter.

### *relational-database-entry-name*

Specify a maximum of 18 characters for the name of the relational database directory entry.

Top

---

## Source journal (SRCJRN)

Specifies the journal on the source system from which the remote journal is being removed. The journal on the source system may be either a local journal or a remote journal.

This is a required parameter.

### Qualifier 1: Source journal

#### *source-journal-name*

Specify the name of the source journal from which the target journal is being removed.

### Qualifier 2: Library

The name of the source journal can be qualified by one of the following library values:

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

#### **\*CURLIB**

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

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

Top

---

## Target journal (TGTJRN)

Specifies the remote journal on the target system.

### Single values

#### **\*SRCJRN**

The target journal name is exactly the same as the source journal name.

### Qualifier 1: Target journal

#### *target-journal-name*

Specify the target journal that is being removed from the source journal.

### Qualifier 2: Library

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

Top

---

## Examples

### Example 1: Removing a Remote Journal Whose Name is the Same as the Source Journal.

```
RMVRMTJRN SRCJRN(RMTLIB/JOURNAL) RDB(CHICAGO)
          TGTJRN(*SRCJRN)
```

This command removes remote journal JOURNAL in library RMTLIB identified by relational database directory entry, CHICAGO, from the source journal JOURNAL in library RMTLIB.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF6981

Remote journal &1 in &2 not removed.

#### CPF6982

Relational database directory entry &1 not valid.

#### CPF6992

Remote journal &1 in &2 not removed.

#### 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.

#### CPF9830

Cannot assign library &1.

Top



---

# Remove Reply List Entry (RMVRPYLE)

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

Parameters  
Examples  
Error messages

The Remove Reply List Entry (RMVRPYLE) command removes an entry from the system reply list. The reply list is used as a source for automatic responses to predefined inquiry messages.

The reply list is only used when an inquiry message is sent by a job that has the inquiry message reply attribute of the system reply list specified; in other words, INQMSGRPY(\*SYSRPLY) was specified. INQMSGRPY(\*SYSRPLY) can be changed with the Change Job (CHGJOB) command.

New entries may be added to the reply list with the Add Reply List Entry (ADDRPYLE) command. Existing entries can be changed with the Change Reply List Entry (CHGRPYLE) command. The entire list of entries can be shown with the Work with System Reply List Entries (WRKRPYLE) display. From the display that is shown by WRKRPYLE, you can add, change, and remove individual entries.

## Restrictions:

1. This command is shipped with public \*EXCLUDE authority and the QPGMR user profile has private authority to use the command.
2. To use this command, you must be signed on as QPGMR, or have all object (\*ALLOBJ) special authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SEQNBR	Sequence number	1-9999, *ALL	Required, Positional 1

Top

---

## Sequence number (SEQNBR)

Specifies the sequence number of the reply list entry being removed from the system reply list.

This is a required parameter.

**\*ALL** Specifies that all reply list entries are removed from the system reply list.

### *sequence-number*

Specify the 4-digit sequence number that specifies the entry that is being removed from the system reply list. Valid values range from 1 through 9999.

Top

---

## Examples

### Example 1: Removing All Entries

```
RMVRPYLE SEQNBR(*ALL)
```

This command removes all entries from the system reply list.

### Example 2: Removing One Entry

```
RMVRPYLE SEQNBR(0001)
```

This command removes from the system reply list the entry that has sequence number 0001.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF2435

System reply list not found.

#### CPF2556

Sequence number &1 not defined in system reply list.

#### CPF2557

System reply list damaged.

#### CPF2558

System reply list currently in use.

[Top](#)

---

# Remove Routing Entry (RMVRTGE)

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

Parameters  
Examples  
Error messages

The Remove Routing Entry (RMVRTGE) command removes a routing entry from the specified subsystem description. The subsystem can be active at the time the command is run.

## Restrictions:

- To use this command, you must have:
  - object operational (\*OBJOPR), object management (\*OBJMGT), and read (\*READ) authority to the specified subsystem description and execute (\*EXECUTE) authority to the library containing the subsystem description.

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>	
SEQNBR	Routing entry sequence number	1-9999	Required, Positional 2

Top

---

## Subsystem description (SBSD)

Specifies the name and library of the subsystem description that contains the routing entry being removed.

This is a required parameter.

### Qualifier 1: Subsystem description

*name* Specify the name of the subsystem description from which the routing entry is being removed.

### Qualifier 2: Library

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

### \*CURLIB

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

*name* Specify the name of the subsystem description's library from which the routing entry is being removed.

Top

---

## Routing entry sequence number (SEQNBR)

Specifies the sequence number of the routing entry being removed. The sequence numbers on the routing entries indicate the order that the routing entries will be processed when comparing values. When a job enters the system, the compare value for the job is compared to the compare value for the routing entries in a subsystem. When the first match is found, the routing information for that entry is used for the new job; therefore, the sequence numbers indicate the compare order.

This is a required parameter.

**1-9999** Specify a sequence number between 1 and 9999.

[Top](#)

---

## Examples

```
RMVRTGE  SBS(OR/PERT)  SEQNBR(9912)
```

This command removes the routing entry 9912 from subsystem description PERT in library OR.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### **CPF1619**

Subsystem description &1 in library &2 damaged.

#### **CPF1691**

Active subsystem description may or may not have changed.

#### **CPF1697**

Subsystem description &1 not changed.

[Top](#)



---

## Remove Search Index Entry (RMVSEARCH)

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

Parameters  
Examples  
Error messages

The Remove Search Index Entry (RMVSEARCH) command removes panel group index entries from a search index.

### Restrictions:

- You must have change (\*CHANGE) authority for the search index and use (\*USE) authority for the library where the search index is located.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SCHIDX	Search index	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Search index	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
PNLGRP	Panel group	<i>Name</i>	Required, Positional 2

Top

---

## Search index (SCHIDX)

Specifies the search index from which the entries are to be removed.

This is a required parameter.

### Qualifier 1: Search index

*name* Specify the name of the search index.

### 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 search index. If no library is specified as the current library for the job, QGPL is used.

*name* Specify the name of the library where the search index is located.

Top

---

## Panel group (PNLGRP)

Specifies the panel group for which entries are to be removed.

This is a required parameter.

*name* Specify the name of the panel group.

[Top](#)

---

## Examples

```
RMVSCHIDX SCHIDX(ACCOUNTING) PNLGRP(PAYROLL)
```

This command removes panel group PAYROLL from search index ACCOUNTING.

[Top](#)

---

## Error messages

None

[Top](#)

---

# Remove Sphere of Control Entry (RMVSOCE)

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

Parameters  
Examples  
Error messages

The Remove Sphere of Control Entry (RMVSOCE) command allows a CL user or program to remove advanced peer-to-peer (APPN) network node control points from the Alert Sphere of Control.

Top

---

## Parameters

Keyword	Description	Choices	Notes
ENTRY	Entry	Values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Network identifier	<i>Communications name</i> , *NETATR	
	Element 2: Control point	<i>Communications name</i>	

Top

---

## Entry (ENTRY)

Specifies the systems to remove from the sphere of control.

You can specify 50 values for this parameter.

### Element 1: Network identifier

#### \*NETATR

Use the NETID network attribute as the value of the Network ID. The NETID network attribute is the value that gets stored. \*NETATR is not stored.

#### *communications-name*

Specify the network ID of the system you want removed from the sphere of control.

### Element 2: Control point

#### *communications-name*

Specify the control point name of the system you want removed from the sphere of control.

Top

---

## Examples

```
RMVSOCE ENTRY((*NETATR RCHSTR1) (*NETATR RCHSTR2))
```

This command removes two systems (RCHSTR1 and RCHSTR2) from the alert sphere of control.

Top

---

## Error messages

Unknown

[Top](#)

---

## Remove Service Table Entry (RMVSRVTBLE)

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

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

The Remove Service Table Entry (RMVSRVTBLE) command is used to remove a service entry from the service table. The service table is used to manage the mapping of network services to ports. You must know the service entry name, the port, and the protocol to remove the entry.

### Restrictions:

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

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
SERVICE	Service	<i>Character value</i>	Required, Positional 1
PORT	Port	1-65535	Required, Positional 2
PROTOCOL	Protocol	<i>Character value</i>	Required, Positional 3

[Top](#)

---

## Service (SERVICE)

Specifies the name of the service entry to be removed.

[Top](#)

---

## Port (PORT)

Specifies the port number assigned to the service to be removed.

[Top](#)

---

## Protocol (PROTOCOL)

Specifies the name of the protocol used by the service to be removed.

[Top](#)

---

## Examples

```
RMVSRVTBLE SERVICE(FTP) PORT(21) PROTOCOL(TCP)
```

This command removes an FTP service entry from the network service table. The service being removed is assigned to port 21 and runs the TCP protocol.

[Top](#)

---

## **Error messages**

### **\*ESCAPE Messages**

#### **TCP2901**

Service entry not found in table.

#### **TCP8050**

\*IOSYSCFG authority required to use &1.

[Top](#)

---

## Remove Server Auth Entry (RMVSVRAUTE)

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

Parameters  
Examples  
Error messages

The Remove Server Authentication Entry (RMVSVRAUTE) command is used to remove server authentication entries from the specified user profile. This authentication information is used by application requesters to connect to application servers. Once an entry is removed, attempting to make new connections to the server will result in either using other server authentication entries or if no matches are found, returning an error to the application.

**Restrictions:** You must have security administrator (\*SECADM) special authority, and object management (\*OBJMGT) and use (\*USE) authorities to the user profile to which the server authentication entry is to be removed, or else be signed on under that user profile, to run this command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	Simple name, *CURRENT	Required, Positional 1
SERVER	Server	Character value, *ALL	Required, Positional 2

Top

---

### User profile (USRPRF)

Specifies the user profile for which the server authentication entry is to be removed.

**\*CURRENT**

The server authentication entry is removed for the current user.

**name** Specify the name of the user profile from which to remove the server authentication entry.

Top

---

### Server (SERVER)

Specifies the name of the application server.

**\*ALL** All server authentication entries for this user profile are to be removed.

**'character-value'**

Specify the name of the application server whose entry is to be removed. Specify no more than 200 characters.

Top

---

## Examples

```
RMVSVRAUTE  USRPRF(*CURRENT)  SERVER('MPLS_RDB')
```

This command removes the server authentication entry for MPLS\_RDB from the current user profile.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### **CPF2204**

User profile &1 not found.

#### **CPF2213**

Not able to allocate user profile &1.

#### **CPF225E**

Server authentication entry does not exist.

#### **CPF226C**

Not authorized to perform function.

[Top](#)



---

## Remove Tape Cartridge (RMVTAPCTG)

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

Parameters  
Examples  
Error messages

The Remove Tape Cartridge (RMVTAPCTG) command removes the specified cartridge identifiers from their current category, or from the category specified, and places them in the eject (\*EJECT) category. The \*EJECT category is not a valid category for input/output (I/O) operations; cartridges in the \*EJECT category cannot be used in the tape device.

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 Other values: <i>Element list</i>	Optional
	Element 1: Category name	<i>Character value</i> , *NOSHARE, *IPL, *NL, *CNV	
	Element 2: Category system	<i>Character value</i> , *CURRENT	
STATION	Station	*DEV, *CNV, *HIGHCAP	Optional

Top

---

### Library device (DEV)

Specifies the library device to be used. The device name must have been previously created 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 the cartridge identifiers that are to be removed.

#### Notes:

- The cartridge identifier should represent the external identifier if the library device has a bar code scanner to read external identifiers.
- If this parameter is specified, the CGY parameter cannot be specified.

#### Single values

**\*ALL** All cartridges are removed.

### 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 to be removed.

Top

---

## Category (CGY)

Specifies the category from which the tape cartridges are to be removed.

**Note:** If the CGY parameter is specified, the CTG parameter cannot be specified.

### Single values

#### **\*SHARE400**

The cartridge identifiers assigned to the \*SHARE400 category are removed. A cartridge in this category can be shared with other systems attached to the same device.

#### **\*INSERT**

All cartridge identifiers assigned to the \*INSERT category are removed. A cartridge in this category has been placed in the library device, but its identifier has not been added to the system.

### Element 1: Category name

#### **\*NOSHARE**

All cartridges identifiers assigned to the \*NOSHARE category are removed. A cartridge with this identifier cannot be shared with other systems.

**\*IPL** The cartridges identifiers assigned to the \*IPL category are removed. A cartridge with this identifier can be used for an alternate IPL.

**\*NL** All cartridges identifiers assigned to the \*NL category are removed. A cartridge with this identifier is used as a non-labeled tape.

**\*CNV** All cartridge identifiers assigned to the \*CNV category are removed. A cartridge in this category would be automatically removed by OS/400 if the cartridge was loaded and an unload was requested by a user.

#### *character-value*

Specify the name of a user-defined category. The cartridge identifiers assigned to the specified user-defined category are removed.

### 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 attribute is used.

\*\*\*\*\* Attention \*\*\*\*\*  
If the system name is changed, the  
categories associated with all tape  
cartridges in library devices are not valid.  
\*\*\*\*\*

**\*CURRENT**

The category belongs to the system currently running the command.

*character-value*

Specify the name of the system to which the category belongs.

Top

---

## Station (STATION)

Specifies the station to receive the cartridges being ejected.

**\*DEV** If the tape library has a convenience I/O station, the cartridges are placed in it. If the tape library does not have a convenience I/O station, the cartridges are placed in the high capacity output station. If the tape library does not have a high capacity output station, the cartridges are placed in a storage slot within the library. The library door must be opened to physically remove the tape cartridges from the high capacity output station or a storage slot within the library.

**\*CNV** The tape cartridge is ejected to the convenience I/O station. The convenience I/O station allows entry and removal of the tape cartridges from the library without opening the library door. If the tape library does not have a convenience I/O station, the cartridges are placed in the high capacity output station. If the tape library does not have a high capacity output station, the cartridges are placed in a storage slot within the library. The library door must be opened to physically remove the tape cartridges from the high capacity output station or a storage slot within the library.

**\*HIGHCAP**

The tape cartridges are ejected to the high capacity output station in the library device. If the tape library does not have a high capacity output station, the cartridges are placed in the convenience I/O station. If the tape library does not have a convenience I/O station, the cartridges are placed in a storage slot within the library. The library door must be opened to physically remove the tape cartridges from the high capacity output station or a storage slot within the library.

Top

---

## Examples

### Example 1: Removing a Single Cartridge to the \*HIGHCAP station

```
RMVTAPCTG  DEV(LIB01)  CTG(VOL4)  STATION(*HIGHCAP)
```

This command removes the cartridge identifier VOL4 from its current category and places it in the \*EJECT category. The cartridge is placed in the high capacity output station. If no high capacity output station is defined the cartridge will be placed in a convenience I/O station. If no convenience I/O is supported the cartridge will remain in a storage slot.

### Example 2: Removing All Cartridges from the \*IPL Category

```
RMVTAPCTG  DEV(LIB01)  CGY(*IPL)  STATION(*CNV)
```

This command removes all cartridge identifiers in the \*IPL category and places them in the \*EJECT category. The cartridges are placed in the convenience I/O station.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF67AF

&6 cartridges not removed

#### CPF6745

Device &1 not a media library device.

[Top](#)

---

## Remove TCP/IP Host Table Entry (RMVTCPHTE)

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

Parameters  
Examples  
Error messages

The Remove TCP/IP Host Table Entry (RMVTCPHTE) command is used to remove an internet address, all of its associated host names, and the associated text description field from the local host table. The local host table is defined to allow one internet address, four host names and one text description field per entry.

See also the following host table commands:

- Add TCP/IP Host Table Entry (ADDTCPHTE) command adds a new entry to the local host table.
- Change TCP/IP Host Table Entry (CHGTCPHTE) command changes one or more host names or the text description field
- Merge TCP/IP Host Table (MRGTCPHTE) command merges host names, internet addresses, and text comment entries from a physical file member into the local host table. A replace option is also provided that allows the entire local host table to be replaced by the host table entries in a user specified physical file member.
- Rename TCP/IP Host Table Entry (RNMTCPHTE) command renames the internet address of a host table entry to another internet address

### Restrictions:

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

Top

---

## Parameters

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

Top

---

## Internet address (INTNETADR)

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

Top

---

## Examples

```
RMVTCPHTE INTNETADR('132.28.71.5')
```

This command removes the host table entry with an internet address of 132.28.71.5. This includes the internet address, all associated host names, and the text description field associated with the entry.

Top

---

## Error messages

### \*ESCAPE Messages

#### TCP1901

Internet address &1 not valid.

#### TCP1902

Internet address &1 not valid.

#### TCP1907

Internet address entry &1 does not exist.

#### TCP1908

Internet address &1 not valid.

#### TCP1929

Host table not available.

#### TCP8050

\*IOSYSCFG authority required to use &1.

#### TCP9999

Internal system error in program &1.

Top

---

## Remove TCP/IP Interface (RMVTCPIFC)

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

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

The Remove TCP/IP Interface (RMVTCPIFC) command is used to remove a Transmission Control Protocol/Internet Protocol (TCP/IP) interface. The interface cannot be active when you submit this command. The interface must be ended using the End TCP/IP Interface (ENDTCPIFC) or End TCP/IP (ENDTCP) commands.

An interface that is required for an existing route or remote system information (RSI) entry cannot be removed.

This command can be used to remove interfaces that have been specified with the Add TCP/IP Interface (ADDTCPIFC) command.

### Restrictions:

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

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	<i>Character value</i>	Optional, Positional 1
ALIASNAME	Alias name	<i>Simple name</i>	Optional

[Top](#)

---

## Internet address (INTNETADR)

Specifies the internet address of an interface that had previously been added to the TCP/IP configuration with the Add TCP/IP Interface (ADDTCPIFC) command. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

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

### *character-value*

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

[Top](#)

---

## Alias name (ALIASNAME)

Specifies the name of an interface that had previously been added to the TCP/IP configuration with the Add TCP/IP Interface (ADDTCPIFC) command. The alias name specified must be for an IPv4 interface. If the alias name is for an IPv6 interface, an error message will be returned.

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

### *simple-name*

Specify the name associated with the interface to be removed.

Top

---

## Examples

### Example 1: Removing a Token Ring Interface

```
RMVTCPIFC INTNETADR('9.5.11.125')
```

This command removes the interface associated with the internet address 9.5.11.125.

### Example 2: Removing a Token Ring Interface using the Alias Name

```
RMVTCPIFC ALIASNAME(TEST_NETWORK)
```

This command removes the interface associated with the alias name TEST\_NETWORK.

Top

---

## Error messages

### \*ESCAPE Messages

#### TCP1D03

&1 member record length not correct.

#### TCP1D04

Error occurred processing member &1 of &2/&3.

#### TCP1901

Internet address &1 not valid.

#### TCP8050

\*IOSYSCFG authority required to use &1.

#### TCP9999

Internal system error in program &1.

Top



---

## Remove TCP/IP Port Restriction (RMVTCPPORT)

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

Parameters  
Examples  
Error messages

The Remove TCP/IP Port Restriction (RMVTCPPORT) command removes a particular user profile from the list of user profiles that are allowed to use a port or range of ports. The removal of the user profile takes effect as soon as that user profile being removed is no longer using the port or any of the ports within the range of ports.

There are two independent sets of ports. One set is for TCP processing and the other is for UDP processing. They are completely independent sets of ports and have no relationship to one another.

### Restrictions:

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

Top

---

## Parameters

Keyword	Description	Choices	Notes
PORT	Range of port values	<i>Element list</i>	Required, Positional 1
	Element 1: Lower value	1-65535, *ALL	
	Element 2: Upper value	1-65535, <u>*ONLY</u>	
PROTOCOL	Protocol	*UDP, *TCP	Required, Positional 2
USRPRF	User profile	<i>Character value</i>	Required, Positional 3

Top

---

## Range of port values (PORT)

Specifies the port number or range of port numbers that are having the user profile removed from the list of user profiles that are allowed to use that port or range of ports. Valid values range from 1 through 65535. However, ports 1 through 1023 are reserved for use by system-supplied TCP/IP applications. If the user specifies ports 1 through 1023, it can affect the operation of those applications.

This is a required parameter.

### Element 1: Lower value

*1-65535*

Specify the port value or the lower port value (in a range) from which you want the user profile removed.

### Element 2: Upper value

### \*ONLY

The port value specified in the lower port value is the only port value that has the user profile removed.

**\*ALL** The port range values that are removed are 1-65535.

### 1-65535

Specify the upper port value (in a range) from which you want the user profile removed.

Top

---

## Protocol (PROTOCOL)

Specifies the transport protocol associated with the port or range of ports having the user profile removed from the list of user profiles that have exclusive use of a port or range of ports.

This is a required parameter.

**\*UDP** The port is a User Datagram Protocol (UDP) transport protocol port.

**\*TCP** The port is a Transmission Control Protocol (TCP) transport protocol port.

Top

---

## User profile (USRPRF)

Specifies the name of the user profile whose profile is to be removed from the list of user profiles that have exclusive use of the port or range of ports.

This is a required parameter.

*name* Specify the name of the user profile that is to be removed.

Top

---

## Examples

### Example 1: Removing Restriction on a Single User Profile

```
RMVTCPPORT PORT(1590) PROTOCOL(*UDP) USRPRF(USER1)
```

This command removes the restriction for the user profile named USER1 for UDP port 1590. This user profile is removed from the list of user profiles that are allowed to use port number 1590.

### Example 2: Removing Restriction on a Range of Ports

```
RMVTCPPORT PORT(1590 2000) PROTOCOL(*TCP) USRPRF(USER2)
```

This command removes the restriction for the user profile named USER2 for TCP ports 1590 through port 2000. This user profile is removed from the list of user profiles that are allowed to use ports 1590 through 2000.

### Example 3: Removing All Ports

```
RMVTCPPORT PORT(*ALL) PROTOCOL(*TCP) USRPRF(USER3)
```

This command removes the restriction for the user profile named USER3 for TCP port 1 through port 65535. The command will complete successfully even if the user did not have a port within this range restricted. This user profile is removed from the list of user profiles that are allowed to use any of the TCP ports.

Top

---

## Error messages

### \*ESCAPE Messages

#### TCP1D03

&1 member record length not correct.

#### TCP1D04

Error occurred processing member &1 of &2/&3.

#### TCP26E2

User profile &1 damaged.

#### TCP26E4

Port restriction action successful, but TCP/IP errors occurred.

#### TCP26E7

You cannot have '\*ALL' and '\*ONLY'.

#### TCP26FA

Port restriction &1 - &2 not removed.

#### TCP26FB

The user profile &1 does not exist

#### TCP26FC

Upper port value must be \*ONLY.

#### TCP26FD

Port &1 not found.

#### TCP26F1

Range of ports not valid.

#### TCP26F2

Removal of port restriction &1 - &2 not valid.

#### TCP8050

\*IOSYSCFG authority required to use &1.

#### TCP9503

File &3 in library &2 not available.

#### TCP9526

User profile &1 not found.

#### TCP9999

Internal system error in program &1.

Top



---

## Remove Point-to-Point Profile (RMVTCPPPTP)

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

Parameters  
Examples  
Error messages

The Remove Point-to-Point TCP/IP Profile (RMVTCPPPTP) command is used to remove a point-to-point connection profile. There could be an output queue (\*OUTQ) object associated with the profile (QUSRSYS/profile-name) and this \*OUTQ would also be removed.

Optionally, all associated communications objects associated with this profile can also be deleted.

### Restrictions:

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

Top

---

## Parameters

Keyword	Description	Choices	Notes
CFGPRF	Configuration profile	<i>Character value</i>	Required, Positional 1
DLTCMNCFG	Delete communications config	<u>*NO</u> , *YES	Optional

Top

---

## Configuration profile (CFGPRF)

Specifies the point-to-point configuration profile to be removed.

This is a required parameter.

### *character-value*

Specify the name of a valid, inactive point-to-point configuration profile to be removed.

Top

---

## Delete communications config (DLTCMNCFG)

Specifies whether you want to delete all communications configuration objects associated with this point-to-point profile. This would include the line, controller and device descriptions defined for use with this profile. Optionally, for some profile types this could also include any network interface descriptions and connection lists associated with the line description being deleted.

\*NO Do not delete any of the communications configuration currently associated with this point-to-point profile.

\*YES Delete the communications configuration associated with this point-to-point profile.

**Note:** If the communications configuration object to be deleted is found to have other point-to-point profiles or line descriptions using it or is not in a 'VARIED OFF' state, it will not be deleted. If a communications object is not deleted a message will be sent indicating the reason.

Top

---

## Examples

### Example 1: Delete a Point-to-Point Profile

```
RMVTCPPPT  CFGPRF(SLIPPROF)
```

This command will remove the point-to-point profile named SLIPPROF. Any communications configuration objects associated with this profile will not be deleted.

### Example 2: Delete Profile and Related Communications Configuration Objects

```
RMVTCPPPT  CFGPRF(ANSPROF)  DLTCMNCFG(*YES)
```

This command will remove the point-to-point profile named ANSPROF and all communications configuration objects associated with the profile. Additionally, if any of the configuration objects are found to be shared with other point-to-point profiles, such as a line description, then the object will **not** be deleted. If an object is not deleted, a diagnostic message will be sent indicating which object was not deleted and why.

Top

---

## Error messages

### \*ESCAPE Messages

TCP83E0

Point-to-point profile &1 not removed.

Top

---

# Remove TCP/IP Remote System (RMVTCPRSI)

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

Parameters  
Examples  
Error messages

The Remove TCP/IP Remote System Information (RMVTCPRSI) command removes a remote system information entry that associates an internet address with an X.25 network address in the TCP/IP configuration.

If you attempt to remove a remote system information entry that is associated with an internet address that is active on an X.25 network's SVC or PVC, the remove operation fails.

## Restrictions:

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

Top

---

## Parameters

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

Top

---

## Internet address (INTNETADR)

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

Top

---

## Examples

```
RMVTCPRSI INTNETADR('128.1.1.10')
```

This command removes the internet address named 128.1.1.10 from the TCP/IP configuration remote system information data, along with its corresponding X.25 network address and other data.

Top

---

## Error messages

### \*ESCAPE Messages

- TCP1D03**  
&1 member record length not correct.
- TCP1D04**  
Error occurred processing member &1 of &2/&3.
- TCP1901**  
Internet address &1 not valid.
- TCP1902**  
Internet address &1 not valid.
- TCP1908**  
Internet address &1 not valid.
- TCP26D4**  
Remote system information not removed.
- TCP26D5**  
Error occurred processing file.
- TCP8050**  
\*IOSYSCFG authority required to use &1.
- TCP9999**  
Internal system error in program &1.

Top



---

## Remove TCP/IP Route (RMVTCPRTE)

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

Parameters  
Examples  
Error messages

The Remove TCP/IP Route (RMVTCPRTE) command removes a previously identified route from the Transmission Control Protocol/Internet Protocol (TCP/IP) configuration.

Five parameter values uniquely define a route. These values are the route destination (RTEDEST), the subnet mask (SUBNETMASK), the type of service (TOS), the internet address of the next system on the route (NEXTHOP), and the preferred binding interface (BINDIFC). For default routes and default multicast routes (\*DFTRROUTE and \*DFTMCAST), the NEXTHOP, TOS and BINDIFC values uniquely define the route because the SUBNETMASK is always \*NONE.

**Note:** When a RMVTCPRTE command is entered using option 4 of the Work with TCP/IP Routes display, a confirmation display is shown. This display warns that the removal of a route might affect active TCP connections and that unpredictable results might occur. A confirmation display is not shown when the RMVTCPRTE CL command is issued directly.

### Restrictions:

- You must have input/output system configuration (\*IOSYSCFG) special authority to run this command.
- Attempts to remove a route that is required to reach an existing RSI entry will fail.

Top

---

## Parameters

Keyword	Description	Choices	Notes
RTEDEST	Route destination	Character value, *DFTRROUTE, *DFTMCAST	Required, Positional 1
SUBNETMASK	Subnet mask	Character value, *NONE, *HOST	Required, Positional 2
TOS	Type of service	*MINDELAY, *MAXTHRPUT, *MAXRLB, *MINCOST, *NORMAL	Optional, Positional 3
NEXTHOP	Next hop	Character value	Optional
BINDIFC	Preferred binding interface	Character value, *NONE	Optional

Top

---

## Route destination (RTEDEST)

Specifies the route destination being removed. You must specify all 4 bytes that make up an internet address though some of the bytes may be equal to 0. For example, a route to all the hosts on the 9.5.11 subnet is identified by entering 9.5.11.0 for the route destination. Used in combination with a subnetmask, type of service value, and next hop, the route destination uniquely identifies a route to a network or system.

### \*DFTRROUTE

Specifies that a default route entry is being removed. A default route entry is used by the system

to route data that is being sent to a remote destination that does not have a specific route defined. The default route entries are used based on the availability of the next hop gateway and the type of service (TOS). If the application requests a specific TOS, the TOS of the default route used must match the TOS requested. If no default route is found that matches the requested TOS, the first available default route with a TOS of \*NORMAL is used.

#### **\*DFTMCAST**

Use the \*DFTMCAST special value to indicate that the static route you are removing is a default Multicast route. The default Multicast route is used by an application when no specific route is specified.

**Note:** When RTEDEST(\*DFTMCAST) is specified, then SUBNETMASK(\*NONE) must also be specified and the NEXTHOP parameter must be a local TCP/IP interface (on this system).

#### *character-value*

Specify the route destination being changed. The route destination can be specified in the form *nnn.0.0.0*, for Class A, *nnn.nnn.0.0* for Class B, and *nnn.nnn.nnn.0* for Class C, or *nnn.nnn.nnn.nnn* for any combination thereof, where *nnn* is a decimal number ranging from 0 through 255.

Any combination thereof means that you may specify a route, such as 9.5.0.0 to the hosts on the 9.5 subnet, even though all 9.5.x.x addresses are class A network addresses.

Exceptions:

- The first byte (octet) must be greater than 0 and less than 255.
- The last byte (octet) may not equal 255.
- The last byte (octet) may not equal 0 if \*HOST is specified for the SUBNETMASK value.
- Routes to a broadcast address are not allowed.

Top

---

## **Subnet mask (SUBNETMASK)**

Specifies a bit mask that identifies to TCP/IP which bits of the value specified for the route destination (RTEDEST) compose the network and subnet portions of the internet address. By defining the network portion and subnetwork portion of the RTEDEST address, the subnet mask also defines which bits of the RTEDEST address make up the host portion. The mask is a 32-bit combination that is logically ANDed with the internet address to determine a particular subnetwork. The bits of the mask set to the value one (1) determine the network and subnetwork portions of the address. The bits set to the value zero (0) determine the host portion of the address.

#### **\*NONE**

No subnet mask is used. A subnet mask is not used when specifying default routes. For example, when RTEDEST(\*DFTMCAST) or RTEDEST(\*DFTRROUTE) is specified, SUBNETMASK(\*NONE) must also be specified.

#### **\*HOST**

The internet address value specified in the route destination field is a host address. The subnetmask value is calculated to be 255.255.255.255.

#### *character-value*

Specify the mask of the subnet field. The internet address is in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. The subnetmask network class's network ID part of the destination route's internet address must equal 255. For example, a destination route's internet address value of 129.35.11.0 is a Class B subnet. The network ID part of its address is 129.35. The upper 2 bytes must designate 255 in the subnetmask. The subnetmask must appear in the format 255.255.x.x, where x is determined by the user. The portion of the subnetmask that is associated with the network portion of a particular class of address must equal 255.

---

## Type of service (TOS)

Specifies the type of service to be used. The type of service defines how the internet hosts and routers should make trade-offs between throughput, delay, reliability, and cost.

\*NORMAL: Normal service is used for delivery of data.

\*MINDELAY: Minimize delay means that prompt delivery is important for data on this connection.

\*MAXTHRPUT: Maximize throughput means that a high data rate is important for data on this connection.

\*MAXRLB: Maximize reliability means that a higher level of effort to ensure delivery is important for data on this connection.

\*MINCOST: Minimize monetary cost means that lower cost is important for data on this connection.

Top

---

## Next hop (NEXTHOP)

Specifies the internet address of the next system (gateway) on the route.

### *character-value*

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

Top

---

## Preferred binding interface (BINDIFC)

Preferred Binding Interface (BINDIFC). Specify the IP interface this route will be bound to. The bind is absolute.

**Note:** This parameter is useful only when there are multiple interfaces defined on your system for the same network. BINDIFC allows you to define which interface should be used to reach the network for a particular route destination. In this way you can distribute traffic (load balancing) across multiple interfaces so all routes do not use the same interface to reach the network.

If the IP interface you specify is active, this route will bind to it, otherwise it will follow the normal route binding rules (which are also used when BINDIFC is defined as \*NONE).

### \*NONE

No particular IP interface will be bound to this route. The first active IP interface on the network defined by the NEXTHOP and SUBNETMASK parameters will be used. This is the default value.

### *character-value*

Specify the internet address (IP address) of the interface you want this route to bind to.

Top

---

## Examples

### Example 1: Removing a Route

```
RMVTCPRTE RTEDEST('132.65.0.0') SUBNETMASK('255.65.0.0')
          TOS(*MINDELAY) NEXTHOP('9.5.15.1')
```

This command removes the route identified as 132.65.0.0 with a subnetmask of 255.65.0.0, a type of service of \*MINDELAY, and a next hop of 9.5.15.1.

### Example 2: Removing a Default Route

```
RMVTCPRTE RTEDEST(*DFTRROUTE) SUBNETMASK(*NONE)
          NEXTHOP('186.34.76.92')
```

This command removes a host route identified as a default route (\*DFTRROUTE). The subnetmask is specified as \*NONE and the type of service defaults to \*NORMAL. The subnetmask, type of service, and next-hop value differentiate this \*DFTRROUTE from the other possible eight \*DFTRROUTE entries.

Top

---

## Error messages

### \*ESCAPE Messages

#### TCP1D03

&1 member record length not correct.

#### TCP1D04

Error occurred processing member &1 of &2/&3.

#### TCP1901

Internet address &1 not valid.

#### TCP261C

Process completed successfully.

#### TCP264A

&2 &1 not removed.

#### TCP8050

\*IOSYSCFG authority required to use &1.

#### TCP9999

Internal system error in program &1.

Top

---

## Remove TCP/IP Server (RMVTCPSVR)

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

Parameters  
Examples  
Error messages

The Remove TCP/IP Server (RMVTCPSVR) command is used to remove a server that was previously added with the Add TCP/IP Server (ADDTCPSVR) command.

### Restrictions:

- You must have input/output system configuration (\*IOSYSCFG) and all object (\*ALLOBJ) special authorities to run this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SVRSPCVL	Server special value	<i>Character value</i>	Required, Positional 1

Top

---

## Server special value (SVRSPCVL)

Specifies the special value of the server that will be removed.

*character-value*

Specify the special value of the server to be removed.

Top

---

## Examples

```
RMVTCPSVR SVRSPCVL(*XYZ)
```

This command removes the server special value \*XYZ from the list of servers supported by the STRTCPSVR (Start TCP/IP Server) and ENDTCPSVR (End TCP/IP Server) CL commands.

Top

---

## Error messages

### \*ESCAPE Messages

#### TCP1641

TCP/IP server &1 not removed.

Top



---

## Remove TCP/IP Table (RMVTCPTBL)

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

Parameters  
Examples  
Error messages

The Remove TCP/IP Table (RMVTCPTBL) command is used to remove ( unload ) from use, any IP Filter Table, IP Network Address Translation Table or both tables from use.

### Restrictions:

You must have \*IOSYSCFG special authority to use this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
TBL	Table	<u>*ALL</u> , *IPFTR, *IPNAT	Optional, Positional 1

Top

---

## Table - Help

Specifies which table to remove from use. Note that, for selected table types, the command will remove them from all physical interfaces.

The possible values are:

**\*ALL** Both the IP Filter rules and the IP Network Address Translation rules will be removed.

**\*IPFTR**  
Only the IP Filter rules will be removed.

**\*IPNAT**  
Only the IP Network Address Translation rules will be removed.

Top

---

## Examples

RMVTCPTBL

This command removes all rule files from use.

Top

---

## Error messages

\*ESCAPE Messages

**TCP8050**

\*IOSYSCFG authority required to use &1.

[Top](#)



---

## Remove Trace (RMVTRC)

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

Parameters  
Examples  
Error messages

The Remove Trace (RMVTRC) command removes all or part of the traces previously specified in one or more Add Trace (ADDTRC) commands for use in debugging the programs. Any trace data already created by the traces being removed is not affected by this command. This data can be removed by the Clear Trace Data (CLRTRCDTA) command. The tracing limits specified in the Change Debug (CHGDBG) or Start Debug (STRDBG) commands are not changed.

On the Remove Trace (RMVTRC) command, the user specifies the HLL statement identifiers or the machine instruction numbers that correspond to the ranges that he no longer wants traced. To remove a trace, exactly the same range as specified on the Add Trace (ADDTRC) command must be specified. A maximum of five sets of trace ranges can be specified in one command.

### 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 remove a trace from a bound program.

Top

---

## Parameters

Keyword	Description	Choices	Notes
STMT	Trace statements to remove	Single values: <b>*ALL</b> Other values (up to 5 repetitions): <i>Element list</i>	Optional, Positional 1
	Element 1: Starting statement identifier	<i>Character value</i>	
	Element 2: Ending statement identifier	<i>Character value</i>	
PGM	Program	<i>Name</i> , <b>*DFTPGM</b> , <b>*ALL</b>	Optional, Positional 2

Top

---

## Trace statements to remove (STMT)

Specifies the statement identifiers or machine instruction numbers of the trace statements that are no longer to be traced. To remove a trace from a program, the same statement identifiers must be specified here that were specified on the Add Trace (ADDTRC) command unless **\*ALL** is specified.

The method used to specify the trace statements on the ADDTRC command (that is, HLL statement identifiers versus machine instruction numbers) must also be used here to remove them.

**\*ALL** All high-level language (HLL) statements and/or machine instructions in the specified programs are no longer traced regardless of how the trace was defined by the Add Trace (ADDTRC) command.

***start-statement-identifier stop-statement-identifier***

Specify the HLL statement identifier (or the machine instruction number) of the first trace statement to be removed and, optionally, the identifier of the last statement to be removed from future tracing. However, if the last statement was specified on the Add Trace (ADDTRC) command, the last statement must also be specified here. As many as five trace ranges can be specified in the program for each use of this command.

The method used to specify the trace statements on the Add Trace (ADDTRC) command (that is, HLL statement identifiers versus machine instruction numbers) must also be used here to remove them.

Top

---

## Program (PGM)

Specifies the program (or all programs) containing the trace statements to be removed from future tracing operations.

**\*DFTPGM**

The program previously specified as the default program contains the statements to be removed from tracing.

**\*ALL** All programs that currently have trace ranges in them have **all** of their trace ranges removed; no tracing can be done in any of the programs in debug mode unless more traces are added by the ADDTRC command. \*ALL is not valid unless no value is supplied for the **Trace statements to remove (STMT)** parameter.

**name** Specify the name of the program that has the specified trace statements (or all trace statements) removed.

Top

---

## Examples

RMVTRC

This command removes all the trace statements used for tracing in the program currently specified as the default program.

Top

---

## Error messages

**\*ESCAPE Messages**

**CPF1999**

Errors occurred on command.

Top

---

# Remove Trace Filter (RMVTRCFTR)

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

Parameters  
Examples  
Error messages

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

## Restrictions:

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

Top

---

## Parameters

Keyword	Description	Choices	Notes
FTR	Filter	<i>Generic name, name</i>	Required, Positional 1

Top

---

## Filter (FTR)

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

This is a required parameter.

### *generic-name*

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

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

Top

---

## Examples

### Example 1: Removing a Single Filter

```
RMVTRCFTR FTR(SAMPLE)
```

This command removes the trace filter named SAMPLE.

### Example 2: Removing All Filters that Start with SAM

```
RMVTRCFTR FTR(SAM*)
```

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

[Top](#)

---

## Error messages

Unknown

[Top](#)

---

## Remove Work Station Entry (RMVWSE)

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

Parameters  
Examples  
Error messages

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

### Restrictions:

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

Top

---

## Parameters

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

Top

---

## Subsystem description (SBSD)

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

This is a required parameter.

### Qualifier 1: Subsystem description

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

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

- QLPINSTALL
- QSYSSBSD

## Qualifier 2: Library

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

### **\*CURLIB**

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

*name* Specify the library where the subsystem description is located.

Top

---

## Work station name (WRKSTN)

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

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

### *generic-name*

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

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

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

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

Top

---

## Work station type (WRKSTNTYPE)

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

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

**3179** 3179 work station.

**3180** 3180 work station.

**3196** 3196 work station.

**3197** 3197 work station.

**3277** 3277 work station.

**3278** 3278 work station.

**3279** 3279 work station.

**3476** 3476 work station.

**3477** 3477 work station.

**3486** 3486 work station.

**3487** 3487 work station.

**5251** 5251 work station.

- 5291** 5291 work station.  
**5292** 5292 color work station.  
**5555** 5555 double-byte character set (DBCS) capable work station.

**\*ASCII**

All ASCII work station device types.

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

**\*CONS**

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

**\*NONASCII**

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

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

Top

---

## Examples

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

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

Top

---

## Error messages

**\*ESCAPE Messages**

**CPF1619**

Subsystem description &1 in library &2 damaged.

**CPF1691**

Active subsystem description may or may not have changed.

**CPF1697**

Subsystem description &1 not changed.

Top





---

## Rename Object (RNM)

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

Parameters  
Examples  
Error messages

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

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

- REN

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

### Restrictions:

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

Top

---

## Parameters

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

---

## Object (OBJ)

Specifies the path name of the object to be renamed.

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Top

---

## New object (NEWOBJ)

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

Top

---

## Examples

### Example 1: Renaming an Object Link

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

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

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFA085

Home directory not found for user &1.

#### CPFA08E

More than one name matches pattern.

#### CPFA093

Name matching pattern not found.

#### CPFA09C

Not authorized to object. Object is &1.

#### CPFA09D

Error occurred in program &1.

#### CPFA0A1

An input or output error occurred.

#### CPFA0A3

Path name resolution causes looping.

#### CPFA0A6

Number of links exceeds maximum allowed for the file system.

**CPFA0A7**

Path name too long.

**CPFA0A9**

Object not found. Object is &1.

**CPFA0AA**

Error occurred while attempting to obtain space.

**CPFA0AB**

Operation failed for object. Object is &1.

**CPFA0B1**

Requested operation not allowed. Access problem.

**CPFA0B2**

No objects satisfy request.

**CPFA0B4**

NEWOBJ parameter cannot start with a slash.

**CPFA0B5**

The NEWOBJ parameter cannot contain path.

**CPFA0C4**

Object not a file. Object is &1.

Top



---

## Rename Directory Entry (RNMDIRE)

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

Parameters  
Examples  
Error messages

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

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

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

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

Top

---

### Parameters

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

Top

---

### Old user identifier (OLDUSRID)

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

This is a required parameter.

#### Element 1: User ID

*character-value*

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

**Element 2: Address**

*character-value*

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

Top

---

## **New user identifier (NEWUSRID)**

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

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

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

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

### **Single values**

**\*BACKOUT**

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

### **Element 1: User ID**

*character-value*

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

### **Element 2: Address**

*character-value*

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

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

Top

---

## Forward from user identifier (FWDFRM)

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

### Single values

#### \*NONE

Distributions are not forwarded.

#### \*OLDUSRID

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

### Element 1: User ID

#### *character-value*

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

### Element 2: Address

#### *character-value*

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

Top

---

## Change network user identifier (NETUSRID)

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

#### \*SAME

The value does not change.

#### \*NEWUSRID

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

Top

---

## Examples

### Example 1: Renaming a User ID

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

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

### Example 2: Renaming a User ID and Network User ID

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

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

---

## Error messages

### \*ESCAPE Messages

#### CPF897F

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

#### CPI90F1

Rename completed with errors. See previously listed messages.



---

## Rename Document Library Object (RNMDLO)

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

Parameters  
Examples  
Error messages

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

### Restrictions:

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

Top

---

## Parameters

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

Top

---

## Document library object (DLO)

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

This is a required parameter.

Top

---

## New document library object (NEWDLO)

Specifies the new name for the document or folder.

This is a required parameter.

Top

---

## In folder (FLR)

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

### \*NONE

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

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

Top

---

## Examples

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

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

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF8A14

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

Top

---

## Rename Distribution List (RNMDSTL)

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

Parameters  
Examples  
Error messages

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

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

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

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

Top

---

### Parameters

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

Top

---

### List identifier (LSTID)

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

The possible list identifier value is:

*list-ID*

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

The possible list qualifier value is:

*list-ID-qualifier*

Specify the list ID qualifier of the distribution list.

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

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

This is a required parameter.

Top

---

## New list identifier (NEWLSTID)

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

The possible list identifier value is:

### *list-ID*

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

The possible list qualifier value is:

### *list-ID-qualifier*

Specify the list ID qualifier of the distribution list.

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

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

This is a required parameter.

Top

---

## Examples

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

This command renames a distribution list that contains the members of Department&#8201;ABC. The list ID is being changed to correspond to a new department name, XYZ. If the new list ID is unique, the distribution list is changed.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9A8B

Distribution list &1 &2 not renamed.

#### CPF9024

System cannot get correct record to finish operation.

#### CPF9845

Error occurred while opening file &1.

#### CPF9846

Error while processing file &1 in library &2.

#### CPF9847

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





---

## Rename LAN Adapter (RNMLANADPI)

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

Parameters  
Examples  
Error messages

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

**Restriction:** Adapter names must be unique.

[Top](#)

---

### Parameters

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

[Top](#)

---

### Adapter (ADPTNAME)

Specifies the name that is currently associated with the adapter.

This is a required parameter.

[Top](#)

---

### New name (NEWNAME)

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

This is a required parameter.

[Top](#)

---

### Examples

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

This command changes the name from ACCTG3A to BILLING2.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF8B48

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

#### CPF8B68

Line description &23 not found.

#### CPF8B74

Request to display active adapters failed.

#### CPF8B75

No adapter entries in network adapter file.

#### CPF8B76

No functional addresses for adapter.

#### CPF8B83

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

Top



---

## Rename Member (RNMM)

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

Parameters  
Examples  
Error messages

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

### Restrictions:

- You must have object management (\*OBJMGT) authority for the file that contains the member that is to be renamed, and have update (\*UPDATE) and execute (\*EXECUTE) authorities for the library in which the file is located.
- This command is conditionally threadsafe. In multithreaded jobs, this command is not threadsafe and fails for Distributed Data Management (DDM) files of type \*SNA.

Top

---

## Parameters

Keyword	Description	Choices	Notes
FILE	Data base file	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Data base file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MBR	Member	<i>Name</i>	Required, Positional 2
NEWMBR	New member	<i>Name</i>	Required, Positional 3

Top

---

## Data base file (FILE)

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

This is a required parameter.

### Qualifier 1: Data base file

*name* Specify the name of the database file.

### Qualifier 2: Library

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

### \*CURLIB

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

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

---

## Member (MBR)

Specifies the physical or logical file member that is to be renamed.

This is a required parameter.

*name* Specify the name of the file member.

---

## New member (NEWMBR)

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

This is a required parameter.

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

---

## Examples

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

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

---

## Error messages

### \*ESCAPE Messages

CPF3178

Member &3 not renamed to &4.

---

## Rename Nickname (RNMNCK)

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

Parameters  
Examples  
Error messages

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

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

### Restrictions:

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

Top

---

## Parameters

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

Top

---

## Nickname (NCK)

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

The possible nickname value is:

*nickname*

Specify the nickname you are renaming.

The possible nickname access values are:

**\*PRIVATE**

The private nickname that you own is being renamed.

**\*PUBLIC**

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

This is a required parameter.

---

## New nickname (NEWNCK)

Specifies the new nickname.

This is a required parameter.

---

## Examples

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

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

---

## Error messages

### \*ESCAPE Messages

#### CPF8AA1

Library QUSRSYS not completely installed.

#### CPF8360

Not enough storage for commitment control operation.

#### CPF9A80

Public nickname &1 in use.

#### CPF9A80

Public nickname &1 in use.

#### CPF9024

System cannot get correct record to finish operation.

#### CPF905C

Error occurred trying to find a translation table.

#### CPF9838

User profile storage limit exceeded.

---

## Rename Object (RNMOBJ)

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

Parameters  
Examples  
Error messages

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

### Restrictions:

1. You must have object management (\*OBJMGT) authority for the object that is to be renamed and have update (\*UPDATE) and execute (\*EXECUTE) authorities for the library in which the object is located.
2. When a library contains one of the following object types, it cannot be renamed: \*CRG, \*DTADCT, \*JRN, \*JRNRCV, \*SQLPKG, and \*SQLUDT.
3. A PL/I program cannot be renamed after it has been created.
4. Configuration objects including controller descriptions, line descriptions, device descriptions, and network interface descriptions must be varied off in order to be renamed.
5. The following objects cannot be renamed:
  - The job's temporary library (QTEMP)
  - The following system libraries (where xxxxx is a 5-digit number):
    - QSYS or QSYSxxxxx
    - QSYS2 or QSYS2xxxxx
    - QRCL or QRCLxxxxx
    - QRECOVERY or QRCYxxxxx
    - SYSIBM or SYSIBxxxxx
    - QQALIB
    - QSYSCGI
  - The system operator message queue (QSYSOPR)
  - All work station user message queues
  - The system log (QHST)
  - The configuration objects (QCTL and QCONSOLE)
  - The configuration lists (QAPPNRMT, QAPPNLCL, QASYNCLC, QRTLPASTR)
  - The Electronic Customer Support configuration objects (QESLINE, QESPAP, QESCTL, QTILINE, QTICTL, QTIDA, QTIDA2, QIADSP, QIAPRT, QQAHOST)
6. A user library cannot be renamed to any of the following (where xxxxx is a 5-digit number):
  - QSYSxxxxx
  - QSYS or QSYS2xxxxx
  - QRCL or QRCLxxxxx
  - QRECOVERY or QRCYxxxxx
  - SYSIBM or SYSIBxxxxx
  - QTEMP
7. When renaming objects of type \*CSI, \*GSS, \*FNTRSC, \*FORMDF, \*OVL, \*PAGDFN, and \*PAGSEG, the new name for the object cannot exceed 8 characters in length.

8. This command is conditionally threadsafe. The following restrictions apply:
  - In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type \*SNA. This command is also not threadsafe and fails for Distributed Data Management (DDM) files of type \*SNA, when SYSTEM(\*RMT) or SYSTEM(\*FILETYPE) is specified.
  - Renaming objects of type \*CFGL, \*CNNL, \*CTLD, \*DEVD, \*LIND or \*NWID will fail in multithreaded jobs.

**NOTES:**

1. References made to the following items may need to be updated by the user after a rename of a configuration object:
  - Connection lists
  - Work station entries
  - Communication entries
  - Display files
  - Printer files
  - Tape files
  - Diskette files
  - ICF files
  - User profiles
  - Job descriptions
  - CL programs
  - QPRTDEV system value
  - Display descriptions referencing it as an auxiliary printer
  - Communication side information (CSI) objects
  - Distributed data management files (APPC device name)
  - Integrated services digital network (ISDN) controller descriptions that refer to a renamed connection list (CNNL)
  - ISDN line descriptions that refer to a renamed CNNL
  - Other configuration objects. For example, lines, controllers, and other devices that refer to the renamed configuration objects
2. References made to the renamed object by the following items are automatically changed by the system after a rename operation. The reference changes reflect the changes made to the renamed configuration objects.
  - QCONSOLE system values
  - message queues associated with display devices
  - System/36 environment device tables
  - output queues associated with the old printer device
  - local work station controllers associated with a twinaxial data link control (TDLC) line
  - TDLC lines associated with the local or remote work station controller

Top

## Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OBJTYPE	Object type	*ALRTBL, *AUTL, *BNDDIR, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CNL, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DTAARA, *DTAQ, *EDTD, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IMGCLG, *IPXD, *JOBQ, *JOBQ, *LIB, *LIND, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *M36, *M36CFG, *NODGRP, *NODL, *NTBD, *NWID, *OUTQ, *OVL, *PAGDFN, *PAGESEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMQR, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SRVPGM, *SSND, *TBL, *USRIDX, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 2
NEWOBJ	New object	<i>Name</i>	Required, Positional 3
ASPDEV	ASP device	<i>Name, *, *CURASPGRP, *SYSBAS</i>	Optional
SYSTEM	System	<i>*LCL, *RMT, *FILETYPE</i>	Optional, Positional 4

Top

## Object (OBJ)

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

This is a required parameter.

### Qualifier 1: Object

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

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(\*) is the only valid value.

### \*CURLIB

The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(\*) is the only valid value.

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

Top

## Object type (OBJTYPE)

Specifies the object type of the object to be renamed.

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

This is a required parameter.

*object-type*

Specify the object type of the object to be renamed.

Top

---

## New object (NEWOBJ)

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

This is a required parameter.

*name* Specify the new name of the object.

Top

---

## ASP device (ASPDEV)

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

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

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

**\*CURASGRP**

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

**\*SYSBAS**

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

*name* Specify the name of the primary or secondary ASP device to be searched to find the library. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Active' or 'Available'. The system ASP (1) and defined basic user ASPs (2-32) will not be searched.

**Note:** To specify a specific auxiliary storage pool (ASP) device name, you must have use (\*USE) authority for each ASP device in the ASP group.

Top



---

## System (SYSTEM)

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

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

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

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

### **\*FILETYPE**

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

Top

---

## Examples

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

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

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPFA030**

Object already in use.

#### **CPFB8ED**

Device description &1 not correct for operation.

#### **CPF0601**

Not allowed to do operation to file &1 in &2.

#### **CPF0602**

File &1 already in library &2.

#### **CPF0605**

Device file &1 in &2 saved with storage freed.

#### **CPF0610**

File &1 in &2 not available.

#### **CPF0678**

Operation not performed for file name &1 in &2.

#### **CPF1763**

Cannot allocate one or more libraries.

#### **CPF180B**

Function &1 not allowed.

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

**CPF2110**  
Library &1 not found.

**CPF2111**  
Library &1 already exists.

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

**CPF2113**  
Cannot allocate library &1.

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

**CPF2132**  
Object &1 already exists in library &2.

**CPF2136**  
Renaming library &1 failed.

**CPF2139**  
Rename of library &1 failed.

**CPF2140**  
Rename of library &1 previously failed.

**CPF2150**  
Object information function failed.

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

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

**CPF2164**  
Rename of library &2 not complete.

**CPF2166**  
Library name &1 not valid.

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

**CPF2176**  
Library &1 damaged.

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

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

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

**CPF2190**  
Not able to do remote delete or rename request.

**CPF22BC**  
Object &1 type &3 is not program defined.

**CPF2451**  
Message queue &1 is allocated to another job.

**CPF2512**  
Operation not allowed for message queue &1.

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

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

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

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

**CPF2696**  
Object &2 type \*&5 not renamed.

**CPF320B**  
Operation was not valid for database file &1.

**CPF3201**  
File &1 in library &2 already exists.

**CPF3202**  
File &1 in library &2 in use.

**CPF3203**  
Cannot allocate object for file &1 in &2.

**CPF322D**  
Operation not done for data base file &1.

**CPF3220**  
Cannot do operation on file &1 in &2.

**CPF323C**  
QRECOVERY library could not be allocated.

**CPF323D**  
User does not have correct authority.

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

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

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

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

**CPF329D**  
Operation not successful for file &1 in library &2.

**CPF3323**  
Job queue &1 in &2 already exists.

**CPF3330**  
Necessary resource not available.

**CPF3353**  
Output queue &1 in &2 already exists.

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

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

**CPF3467**  
Output queue &1 deleted and then created again.

**CPF3469**  
Operation not allowed for output queue.

**CPF5702**  
File either not DDM file or not found.

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

**CPF8D05**  
Library &1 already exists.

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

**CPF9801**  
Object &2 in library &3 not found.

**CPF9803**  
Cannot allocate object &2 in library &3.

**CPF9807**  
One or more libraries in library list deleted.

**CPF9808**  
Cannot allocate one or more libraries on library list.

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

**CPF9814**  
Device &1 not found.

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

**CPF9830**  
Cannot assign library &1.

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

**CPF9876**  
Protected library &2 cannot be modified.

Top

---

## Rename TCP/IP Host Table Entry (RNMTCPHTE)

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

Parameters  
Examples  
Error messages

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

The local host table is defined to allow 1 internet address, 4 host names and 1 text description field per entry.

See also the following host table commands:

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

### Restrictions:

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

Top

---

## Parameters

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

Top

---

## Internet address (INTNETADR)

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

Top

---

## New internet address (NEWINTNETA)

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

Top

---

## Examples

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

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

Top

---

## Error messages

### \*ESCAPE Messages

#### TCP1901

Internet address &1 not valid.

#### TCP1902

Internet address &1 not valid.

#### TCP1904

Duplicate internet address &1 found in host table.

#### TCP1907

Internet address entry &1 does not exist.

#### TCP1908

Internet address &1 not valid.

#### TCP1910

LOOPBACK internet address &1 not valid.

#### TCP1929

Host table not available.

#### TCP8050

\*IOSYSCFG authority required to use &1.

Top

---

## Rollback (ROLLBACK)

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

Parameters  
Examples  
Error messages

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

When the ROLLBACK command is issued:

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

More information on 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

ROLLBACK

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

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF83D0

Commitment operation not allowed.

**CPF835F**

Commit or rollback operation failed.

**CPF8350**

Commitment definition not found.

**CPF8359**

Rollback operation failed.

**CPF8367**

Cannot perform commitment control operation.

**\*STATUS Messages****CPF83E6**

Commitment control operation completed with resynchronization in progress.

Top



---

# Start RPC Binder Daemon (RPCBIND)

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

Parameters  
Examples  
Error messages

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

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

- STRNFSSVR SERVER(\*RPC)

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

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

QNFSRPCD The RPCBind daemon

## Restrictions

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

Top

---

## Parameters

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

Top

---

## Retrieve RPC registration (RTVRPCREG)

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

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

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

Top

---

## Examples

### Example 1: Start RPC Binder Daemon

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

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

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPFA1B8

\*IOSYSCFG authority required to use &1.

[Top](#)

---

## Convert RPC Source (RPCGEN)

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

Parameters  
Examples  
Error messages

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

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

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

- CVTRPCSRC

### Restrictions:

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

Top

---

## Parameters

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

Top

---

## From file (FROMFILE)

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

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Top

---

## Option (OPTION)

Specifies the compile options.

### \*NOSAMP

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

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

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

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

\***CLTSTUB**

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

\***SVRSTUB**

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

\***CLTSAMP**

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

\***SVRSAMP**

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

Top

---

## Protocol (PROTOCOL)

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

\***NONE**

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

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

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

Top

---

## To file (TOFILE)

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

- filename.h for a header file
- filename\_xdr.c for an XDR file
- filename\_clnt.c for client-side stubs
- filename\_svc.c for server-side stubs
- filename\_client.c for client-side sample files
- filename\_server.c for server-side sample files

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

*'to-file-path name'*

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

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Top

---

## Examples

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

### Example 1: Convert RPC Source to Default Files

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

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

- myrpc.h for a header file
- myrpc\_xdr.c for an XDR file
- myrpc\_clnt.c for client-side stubs
- myrpc\_svc.c for server-side stubs

### Example 2: Convert RPC Source to Client Stubs Only

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

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

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFB416

CVTRPCSRC or RPCGEN command failed.

Top



## Replace Document (RPLDOC)

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

Parameters  
 Examples  
 Error messages

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

### Restrictions:

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

Top

## Parameters

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

---

## Document (TODOC)

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

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

### \*DOCID

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

---

## In folder (TOFLR)

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

### \*NONE

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

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

---

## Document file (DOCFILE)

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

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

### Qualifier 2: Library

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

### \*CURLIB

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

*name* Specify the library where the database file is located.

---

## Document member (DOCMBR)

Specifies the document database file member that is used.

### \*FIRST

The first member created in the database file is used.



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

Top

---

## Document part (DOCPART)

Specifies the part of the document to be replaced.

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

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

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

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

Top

---

## User identifier (USRID)

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

### Single values

**\*CURRENT**  
You are performing the request for yourself.

### Element 1: User ID

#### *character*

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

### Element 2: Address

#### *character*

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

Top

---

## Document identifier (DOCID)

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

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

YYYYMMDDHHMNSSHSSNSNSNSN

where:

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

**\*NONE**

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

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

Top

---

## Document type (DOCTYPE)

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

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

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

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

2-65,535

Specify a document type identifier value. The numbers from 2 through 32,767 are controlled by registering them with the IBM Document Interchange Architecture and are used for IBM-defined document types. The numbers ranging from 32,768 through 65,535 are not registered with IBM and can be used for non-IBM-defined document types.

Top

---

## Document character identifier (DOCCHRID)

Specifies the character identifier (graphic character set and code page) for the document data being used. The character identifier is related to the display device used to create the document data.

### Single values

**\*SYSVAL**

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

**\*DEV D**

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

### Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

## Element 2: Code page

1-32767

Specify the code page to use.

Top

---

## Command character identifier (CMDCHRID)

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

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

### Single values

#### \*SYSVAL

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

#### \*DEV D

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

## Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

## Element 2: Code page

1-32767

Specify the code page to use.

Top

---

## Examples

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

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

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF900B

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

**CPF900C**

Sign on and verify of user failed.

**CPF905C**

Error occurred trying to find a translation table.

**CPF906A**

Document not replaced in library.

**CPF9096**

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

**CPF9847**

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

[Top](#)

---

# Request Order Assistance (RQSORDAST)

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

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

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

- Software upgrades
- Hardware upgrades
- AS/400 information
- Service offerings
- General help (for example, network planning)

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

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

The order information file is sent with all requests.

## Restrictions:

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

There are no parameters for this command.

[Top](#)

---

## Parameters

None

[Top](#)

---

## Examples

RQSORDAST

This command displays the Request Order Assistance entry panel.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Reroute Job (RRTJOB)

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

Parameters  
Examples  
Error messages

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

### Notes:

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

### Restrictions:

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

Top

---

## Parameters

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

Top

---

## Routing data (RTGDTA)

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

### QCMDI

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

### \*RQSDTA

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

*character-value*

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

Top

---

## Request data or command (RQSDTA)

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

**\*NONE**

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

**\*RTGDTA**

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

*character-value*

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

Top

---

## Examples

```
RRTJOB RTGDTA(INQUIRY)
```

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

Top

---

## Error messages

**\*ESCAPE Messages**

**CPF1315**

Command &1 not allowed in this environment.

Top



---

## Resume Breakpoint (RSMBKP)

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

Parameters  
Examples  
Error messages

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

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

### Restrictions:

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

There are no parameters for this command.

[Top](#)

---

## Parameters

None

[Top](#)

---

## Examples

RSMBKP

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

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF1999

Errors occurred on command.

[Top](#)



---

## Resume Controller Recovery (RSMCTRLRCY)

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

Parameters  
Examples  
Error messages

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

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

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

Top

---

### Parameters

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

Top

---

### Controller (CTL)

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

#### CTL Value

##### Controller

**5251** Display station

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

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

#### **\*BSCT**

BSC device (Multipoint tributary and 3270 Device Emulation)

#### **\*APPC**

Advanced program-to-program communications

**\*WSC** Local work station

#### **\*WSCE**

Local work station (extended)

This is a required parameter.

---

## Examples

RSMCTLR CY CTL(TROLL3)

This command resumes error recovery procedures for the controller TROLL3.

---

## Error messages

### \*ESCAPE Messages

**CPF2703**

Controller description &1 not found.

**CPF5924**

Controller &1 does not allow automatic error recovery.

**CPF5928**

Controller &1 not varied on.

**CPF5929**

Controller &1 assigned to another job.

**CPF5930**

Recovery not resumed for controller &1.

**CPF5931**

Recovery not resumed for controller &1.

**CPF5935**

Error occurred during command processing.

**CPF5936**

Not authorized to controller &1.

---

## Resume Device Recovery (RSMDEVRCY)

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

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

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

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

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

[Top](#)

---

### Parameters

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

[Top](#)

---

### Device (DEV)

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

#### DEV Type

##### Device

5219 Printer (work station)

5224 Printer (work station)

5225 Printer (work station)

5251 Display station

5252 Dual display station

5256 Printer (work station)

5291 Display station

5292 Display station

\*PLU1 Physical unit (type 1)

\*BSC BSC device

#### \*BSCT

BSC multipoint tributary

### \*APPC

Advanced program-to-program communications

This is a required parameter.

[Top](#)

---

## Examples

```
RSMDEVRCY  DEV(WSPR03)
```

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

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF5923

Device &1 does not allow automatic error recovery.

#### CPF5925

Device &1 not varied on.

#### CPF5926

Recovery not resumed for device &1.

#### CPF5927

Recovery not resumed for device &1.

#### CPF5935

Error occurred during command processing.

#### CPF9814

Device &1 not found.

#### CPF9825

Not authorized to device &1.

[Top](#)

---

# Resume Line Recovery (RSMLINRCY)

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

Parameters  
Examples  
Error messages

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

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

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

Top

---

## Parameters

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

Top

---

## Line (LINE)

Specifies the name of the communications line whose recovery is started again.

This is a required parameter.

Top

---

## Examples

```
RSMLINRCY LINE(NYC2)
```

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

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2704

Line description &1 not found.

**CPF5917**

Not authorized to line description &1.

**CPF5932**

Cannot access line &1.

**CPF5933**

Line &1 not varied on.

**CPF5934**

Recovery not resumed for line &1.

**CPF5935**

Error occurred during command processing.

[Top](#)



---

# Resume NWI Recovery (RSMNWIRCY)

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

Parameters  
Examples  
Error messages

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

Top

---

## Parameters

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

Top

---

## Network interface description (NWI)

Specifies the name of the network interface description whose automatic error recovery is restarted.

This is a required parameter.

Top

---

## Examples

```
RSMNWIRCY  NWID(ISDNNET)
```

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

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF591A

Not authorized to network interface description &1.

#### CPF593A

Network interface &1 not varied on.

#### CPF593B

Network interface description &1 not found.

**CPF593C**

Cannot access network interface &1.

**CPF593D**

Recovery not resumed for network interface &1.

[Top](#)

## Restore Object (RST)

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

Parameters  
 Examples  
 Error messages

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

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

### Restrictions:

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

Top

## Parameters

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

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

Top

---

## Device (DEV)

Specifies the device from which the objects are restored.

For more information on specifying device path names, refer to "Specifying the device name" in the Backup and Recovery information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.

This is a required parameter.

### *'save-file-path-name'*

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

### *'optical-device-path-name'*

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

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

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

### *'tape-device-path-name'*

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

### *'media-definition-path-name'*

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

For information about creating a media definition, see the Create Media Definition API in the System API Reference information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.

For information about using a media definition, see the Backup and Recovery book, SC41-5304 and the Backup Your Server topic in the System API Reference information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.

Top

---

## Objects (OBJ)

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

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.

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

#### Element 1: Name

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

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

#### *path-name*

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

#### Element 2: Include or omit

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

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

#### \*INCLUDE

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

#### \*OMIT

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

#### Element 3: New object name

Specifies the new path name of the object.

#### \*SAME

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

#### *path-name*

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

Top

---

## Name pattern (PATTERN)

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

#### Element 1: Pattern

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

#### *character-value*

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

## Element 2: Include or omit

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

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

### \*INCLUDE

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

### \*OMIT

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

Top

---

## Directory subtree (SUBTREE)

Specifies whether directory subtrees are included in the restore operation.

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

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

### \*NONE

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

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

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

Top

---

## Output (OUTPUT)

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

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

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

### \*NONE

No output is created.

### \*PRINT

Output information about the restore will be printed.

### *'stream-file-path-name'*

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

*'user-space-path-name'*

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

Top

---

## Create parent directories (CRTPRNDIR)

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

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

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

Top

---

## Parent directory owner (PRNDIROWN)

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

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

### **\*PARENT**

The owner of a parent directory being created by the restore will be the same as the owner of the directory it is being created into. For example, if object *'/a/b/c/file1'* is being restored and directory *'/a'* exists but the *'/b'* and *'/b/c'* directories do not exist, the *'/b'* and *'/b/c'* directories are created with the same owner as the *'/a'* directory.

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

Top

---

## Volume identifier (VOL)

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

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

### Single values

#### **\*MOUNTED**

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

parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

### Other values (up to 75 repetitions)

#### *character-value*

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

Top

---

## Label (LABEL)

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

### \*SEARCH

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

#### *character-value*

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

Top

---

## Sequence number (SEQNBR)

Specifies the tape file sequence number to be used.

### \*SEARCH

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

#### *1-16777215*

Specify the sequence number of the file.

Top

---

## End of media option (ENDOPT)

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

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

### \*REWIND

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

### \*LEAVE

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

### \*UNLOAD

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

Top



---

## Optical file (OPTFILE)

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

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

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

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

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

**'optical-file-path-name'**

Specify the path name of the optical file.

Top

---

## Type of output information (INFTYPE)

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

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

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

**\*SUMMARY**

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

Top

---

## System (SYSTEM)

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

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

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

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

Top

---

## Date when saved (SAVDATE)

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

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

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

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

Top

---

## Time when saved (SAVTIME)

Specifies the time the objects were saved.

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

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

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

### Notes:

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

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

Top

---

## Option (OPTION)

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

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

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

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

Top

---

## Allow object differences (ALWOBJDIF)

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

### Notes:

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

The types of differences include:

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

### Single values

#### **\*NONE**

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

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

### Other values (up to 3 repetitions)

#### **\*AUTL**

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

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

#### **\*OWNER**

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

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

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

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

Top

---

## Force object conversion (FRCOBJCVN)

Specifies whether to convert user objects to the format required for use in the current version of the operating system when the objects are restored.

### Notes:

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

### Single values

#### \*SYSVAL

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

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

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

### Element 1: Convert during restore

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

#### Notes:

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

### Element 2: Objects to convert

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

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

Top

---

## Object ID (OBJID)

This parameter has been disabled and is no longer valid.

Top

---

## Examples

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

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

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

### Example 2: Restoring a Library

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

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

### Example 3: Restoring All Files in MYLIB

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

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

### Example 4: Restoring All Objects in the Current Directory

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

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

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

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

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

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

### Example 5: Omitting Objects

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

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

### Example 6: Renaming or Moving Objects

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

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

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

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

### Example 7: Restoring From a Save File

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

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

### Example 8: Using Symbolic Links

Assume the current directory contains the following symbolic links.

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

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

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

To restore the objects in SomeDirectory from device TAP01:

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

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFA0DB

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

#### CPFA0DC

Object not a QDLS object. Object is &1.

#### CPF370C

Not authorized to ALWOBJDIF parameter.

#### CPF3707

Save file &1 in &2 contains no data.

#### CPF3727

Duplicate device &1 specified on device name list.

#### CPF3738

Device &1 used for save or restore is damaged.

#### CPF3743

File cannot be restored, displayed, or listed.

#### CPF3768

Device &1 not valid for command.

#### CPF3782

File &1 in &2 not a save file.

#### CPF3794

Save or restore operation ended unsuccessfully.

#### CPF380D

Save or restore of entire system completed unsuccessfully.

#### CPF3805

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

#### CPF381E

Not authorized to ALWOBJDIF parameter.

#### CPF3812

Save file &1 in &2 in use.

**CPF382A**  
Specified parameter not valid for QDLS file system.

**CPF382B**  
Parameters not valid with multiple file systems.

**CPF382C**  
OBJ parameter value not valid for QSYS file system.

**CPF382D**  
Specified parameter not valid for QSYS file system.

**CPF382F**  
OBJ parameter value not valid for QDLS file system.

**CPF3823**  
No objects saved or restored.

**CPF3826**  
\*INCLUDE object required on OBJ parameter.

**CPF3828**  
Error occurred while attempting to use &1.

**CPF383A**  
Save or restore ended unsuccessfully.

**CPF383B**  
End of file &1.

**CPF383C**  
Storage limit exceeded for user profile &1.

**CPF383D**  
Cannot use &1.

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

**CPF3833**  
Specified value on DEV parameter not valid.

**CPF3834**  
Too many values specified on the DEV parameter.

**CPF3835**  
Tape devices do not support same densities.

**CPF3839**  
&1 objects restored. &2 not restored.

**CPF384A**  
Volume identifier &1 not valid.

**CPF384B**  
Optical file specified not valid.

**CPF384C**  
Error occurred during CCSID conversion.

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

**CPF3840**  
Specified file for restore operation not found.

**CPF38A5**

Error on the PATTERN parameter.

**CPF5729**

Not able to allocate object &1.

**CPF9802**

Not authorized to object &2 in &3.

**CPF9825**

Not authorized to device &1.

**OPT1498**

Volume name list exhausted on device &1.

**OPT1502**

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

**OPT1605**

Media or device error occurred.

[Top](#)



---

## Restore Authority (RSTAUT)

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

Parameters  
Examples  
Error messages

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

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

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

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

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

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

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

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

**Restrictions:**

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

Top

## Parameters

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

Top

## User profile (USRPRF)

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

**Single values**

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

**Other values (up to 300 repetitions)**

*generic-name*

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

*name*

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

Top

## Saved from ASP device (SAVASPDEV)

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

\*ALLAVL

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

\*

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

### \*SYSBAS

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

### \*CURASPGRP

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

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

Top

---

## Restore to ASP device (RSTASPDEV)

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

### \*SAVASPDEV

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

### \*SYSBAS

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

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

Top

---

## Examples

### Example 1: Restore All Authorities

```
RSTAUT
```

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

### Example 2: Restore Authorities for Specific Users

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

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

### Example 3: Restore Authorities for Specific Libraries

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

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

### Example 4: Restore Authorities After Multiple RSTUSRPRF Commands

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

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

### Example 5: Restore Libraries and Authorities to an Independent ASP

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

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

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2206

User needs authority to do requested function on object.

#### CPF222E

&1 special authority is required.

#### CPF3776

Not all user profiles had all authorities restored.

#### CPF3785

Not all subsystems ended.

#### CPF3855

RSTAUT not allowed at this time.

#### CPF386D

Prestart job failed.

#### CPF9814

Device &1 not found.

#### CPF9833

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

#### CPFB8ED

Device description &1 not correct for operation.

Top

---

## Restore Configuration (RSTCFG)

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

Parameters  
Examples  
Error messages

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

### Restrictions:

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

Top

---

## Parameters

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

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

Top

## Objects (OBJ)

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

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

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

This is a required parameter.

### Single values

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

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

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

### Other values (up to 300 repetitions)

#### *generic-name*

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

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

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

Top

---

## Device (DEV)

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

This is a required parameter.

### Single values

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

### Other values

#### *optical-device-name*

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

#### *tape-media-library-device-name*

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

#### *tape-device-name*

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

Top

---

## Object types (OBJTYPE)

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

### Single values

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

### Other values (up to 10 repetitions)

### *character-value*

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

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

**\*CFGL**

Configuration lists

**\*CNL**

Connection list

**\*COSD**

Class-of-service description

**\*CTLD**

Controller description

**\*DEVD**

Device description

**\*IPXD** Internet Packet Exchange description

**\*LIND**

Line description

**\*MODD**

Mode description

**\*NTBD**

NetBIOS description

**\*NWID**

Network interface description

**\*NWSD**

Network server description

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

Top

---

## Volume identifier (VOL)

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

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

### Single values

**\*MOUNTED**

The objects are restored from the volumes placed in the device specified for the **Device (DEV)** parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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



## Other values (up to 75 repetitions)

### *character-value*

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

Top

---

## Sequence number (SEQNBR)

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

### \*SEARCH

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

### *1-16777215*

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

Top

---

## End of media option (ENDOPT)

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

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

### \*REWIND

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

### \*LEAVE

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

### \*UNLOAD

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

Top

---

## Save file (SAVF)

Specifies the save file used to restore the data.

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

### Qualifier 1: Save file

*name* Specify the name of save file to be used.

### Qualifier 2: Library

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

### **\*CURLIB**

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

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

Top

---

## **System Resource Management (SRM)**

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

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

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

### **\*NONE**

No system resource management information is restored.

### **\*HDW**

All hardware information is restored.

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

Top

---

## **Optical file (OPTFILE)**

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

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

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

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

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

Top

---

## **Allow object differences (ALWOBJDIF)**

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

### **Notes:**

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

The types of differences include:

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

### Single values

#### **\*NONE**

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

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

### Other values (up to 3 repetitions)

#### **\*AUTL**

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

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

#### **\*OWNER**

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

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

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

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

Top

---

## Output (OUTPUT)

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

### **\*NONE**

No output is created.

### **\*PRINT**

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

### **\*OUTFILE**

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

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

Top

---

## **File to receive output (OUTFILE)**

Specifies the database file to which the information is directed when \*OUTFILE is specified for the **Output (OUTPUT)** parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QASRRSTO in QSYS with the format name QSRRST as a model.

### **Qualifier 1: File to receive output**

*name* Specify the name of the database file to which output from the command is directed. If this file does not exist, it is created in the specified library.

### **Qualifier 2: Library**

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

### **\*CURLIB**

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

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

Top

---

## **Output member options (OUTMBR)**

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

### **Element 1: Member to receive output**

#### **\*FIRST**

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

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

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

### **Element 2: Replace or add records**

### \*REPLACE

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

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

Top

---

## Objects to omit (OMITOBJ)

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

### Element 1: Object

#### Qualifier 1: Object

##### \*NONE

No objects are excluded from the operation.

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

##### *generic-name*

Specify the generic name of the objects to be excluded.

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

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

### Element 2: Object type

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

##### *character-value*

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

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

Top

---

## Examples

### Example 1: Restoring All Objects

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

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

### Example 2: Restoring a Device Description

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

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

### Example 3: Restoring a Network Server Description

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

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

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF26C1

Network server &1 not varied off.

#### CPF370C

Not authorized to ALWOBJDIF parameter.

#### CPF3707

Save file &1 in &2 contains no data.

#### CPF3709

Tape devices do not support same densities.

#### CPF3727

Duplicate device &1 specified on device name list.

#### CPF3728

Device &1 specified with other devices.

#### CPF3733

&2 &1 in &3 previously damaged.

#### CPF3738

Device &1 used for save or restore is damaged.

#### CPF3743

File cannot be restored, displayed, or listed.

#### CPF3748

Object information for library &1 damaged.

#### CPF376B

File &1 not found.

#### CPF3767

Device &1 not found.

#### CPF3768

Device &1 not valid for command.

#### CPF3769

File found on media not save/restore file.

#### CPF377B

No objects restored. &3 objects excluded.

#### CPF377C

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

#### CPF3780

Specified file for library &1 not found.

- CPF3782**  
File &1 in &2 not a save file.
- CPF379B**  
Objects from save file &1 in &2 not restored.
- CPF3791**  
While processing &2 &1 in &3, encountered end of file &4.
- CPF3793**  
Machine or ASP storage limit reached.
- CPF3794**  
Save or restore operation ended unsuccessfully.
- CPF3796**  
Storage limit exceeded for user profile &4.
- CPF3805**  
Objects from save file &1 in &2 not restored.
- CPF3807**  
Data decompression error for save file &1 in &2.
- CPF3812**  
Save file &1 in &2 in use.
- CPF3883**  
S/38 device configuration objects not restored.
- CPF5729**  
Not able to allocate object &1.
- CPF9809**  
Library &1 cannot be accessed.
- CPF9812**  
File &1 in library &2 not found.
- CPF9820**  
Not authorized to use library &1.
- CPF9822**  
Not authorized to file &1 in library &2.
- CPF9825**  
Not authorized to device &1.

Top





---

## Restore Document Lib Object (RSTDLO)

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

Parameters  
Examples  
Error messages

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

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

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

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

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

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

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

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

If a document is restored that had a mail log entry when it was saved, the mail log entry is restored if the distribution tracking object exists on the system. If the distribution tracking object does not exist on the system, a message is sent saying that the document was restored without a mail log entry.

If this command ends abnormally, objects are left on the system in an unknown state and cannot be found in a library. This can happen if a power failure occurs when this command is run. The Reclaim Storage (RCLSTG) command can be used to clean up the auxiliary storage and delete most of those objects from the system; however, unknown mail objects are not cleaned up with the RCLSTG command.

When a set of documents and folders are restored, all documents and folders in the set must exist in the same tape, optical volume, or save file.

If a document exists in more than one tape file, the user can control which document is restored by specifying the media file using the sequence number or label parameter. If more than one version of the document exists, the SAVDATE and SAVTIME parameters can also be used to select the correct document.

When text search services are on the system and the user restores a document library object, the text search index for the object is restored.

#### Restrictions:

- This command is shipped with no public authority (\*EXCLUDE).
- You must have save system (\*SAVSYS) or all object (\*ALLOBJ) special authority or be enrolled in the system distribution directory to run this command.
- This command cannot be run when RCLDLO DLO(\*ALL) is running because RCLDLO requires exclusive use of internal objects.
- When saving or restoring to an existing database file using the OUTFILE parameter, you must have execute (\*EXECUTE) authority to the library where the output database file is located.

Top

## Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Single values: *ALL, *SYSOBJNAM, *MAIL Other values (up to 300 repetitions): <i>Character value</i>	Required, Positional 1
SAVFLR	Saved folder	Single values: *ANY, *NONE Other values (up to 300 repetitions): <i>Character value</i>	Optional
RENAME	New object name	Single values: *SAME Other values (up to 300 repetitions): <i>Character value</i>	Optional
RSTFLR	Restore into folder	<i>Character value</i> , *SAME	Optional
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 2
SYSOBJNAM	System object name	Values (up to 300 repetitions): <i>Name</i> , *NONE	Optional
VOL	Volume identifier	Single values: *MOUNTED Other values (up to 75 repetitions): <i>Character value</i>	Optional
SEQNBR	Sequence number	Single values: *SEARCH Other values: <i>Element list</i>	Optional
	Element 1: Beginning sequence number	1-16777215	
	Element 2: Ending sequence number	1-16777215, *ONLY	
ENDOPT	End of media option	*REWIND, *LEAVE, *UNLOAD	Optional
LABEL	Label	<i>Character value</i> , *GEN	Optional
SAVF	Save file	<i>Qualified object name</i>	Optional
	Qualifier 1: Save file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
OPTFILE	Optical file	<i>Path name</i> , *	Optional
NEWOBJ	Object name generation	*SAME, *NEW	Optional
SAVDATE	Date when saved	<i>Date</i>	Optional
SAVTIME	Time when saved	<i>Time</i>	Optional
ALWOBJDIF	Allow object differences	*NONE, *ALL	Optional
SAVASP	Saved from ASP number	1-32, *ANY	Optional

Keyword	Description	Choices	Notes
RSTASP	Restore to ASP number	1-32, <u>*SAVASP</u>	Optional
OUTPUT	Output	<u>*NONE</u> , *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , <u>*FIRST</u>	
	Element 2: Replace or add records	<u>*REPLACE</u> , *ADD	

Top

---

## Document library object (DLO)

Specifies the document library objects to be restored.

This is a required parameter.

### Single values

**\*ALL** All documents, folders, and distribution objects (mail) that are saved on media and meet the criteria specified for the **Saved folder (SAVFLR)** parameter are restored.

**\*MAIL**  
All distribution objects and documents that were referred to by a mail log are restored.

**\*SYSOBJNAM**  
All documents with the system object names specified for the **System object name (SYSOBJNAM)** parameter are restored.

### Other values (up to 300 repetitions)

#### *document-name*

Specify the user-assigned names of the documents to be restored. All documents named must be in the folder specified for the SAVFLR parameter.

Top

---

## Saved folder (SAVFLR)

Specifies the name of the folder on the media from which the documents and folders are restored.

### Single values

**\*ANY** All document library objects that meet the values specified for the **Document library object (DLO)** parameter are restored, regardless of the folders (if any) from which they were saved. This value is valid only if \*ALL, \*MAIL, or \*SYSOBJNAM is specified for the DLO parameter.

**\*NONE**  
All documents that were saved as documents not in any folder are restored. \*NONE is valid for documents only and is valid only when \*ALL is specified for the DLO parameter.

### Other values (up to 300 repetitions)

### *folder-name*

Specify the name of the saved folder from which documents are to be restored, or the name of the folders to be restored if \*ALL was specified for the DLO parameter. A maximum of 63 characters can be specified for the folder name. When \*ALL is specified for the DLO parameter, a maximum of 300 folder names can be specified. The name of a saved folder must be specified when DLO(document-name) is specified.

Top

---

## **New object name (RENAME)**

Specifies the new user-assigned name for the restored document.

### **Single values**

#### \*SAME

The documents are restored with the names they had when they were saved.

### **Other values (up to 300 repetitions)**

#### *document-name*

Specify the new user-assigned names that the documents have after being restored. When document names are specified for the **Document library object (DLO)** parameter, a maximum of 300 user-assigned names can be specified for documents being restored.

Top

---

## **Restore into folder (RSTFLR)**

Specifies the name of the folder in which the restored folders and documents to be restored will be placed. The folder must exist on the system or when \*ALL is specified for the **Document library object (DLO)** parameter, the saved folder must exist on the media.

#### \*SAME

The folders and documents to be restored are placed into the same folder from which they were saved.

### *folder-name*

Specify the name of the folder where the restored documents and folders are to be placed.

Top

---

## **Device (DEV)**

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

This is a required parameter.

### **Single values**

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

### **Other values**

#### *optical-device-name*

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

*tape-media-library-device-name*

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

*tape-device-name*

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

Top

---

## System object name (SYSOBJNAM)

Specifies the system object names of the documents to be restored if \*SYSOBJNAM is specified for the **Document library object (DLO)** parameter.

### Single values

\*NONE

A system object name is not specified.

### Other values (up to 300 repetitions)

*system-object-name*

Specify the system object names of the documents to be restored.

Top

---

## Volume identifier (VOL)

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

### Single values

\*MOUNTED

The objects are restored from the volumes placed in the device specified for the **Device (DEV)** parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

### Other values (up to 75 repetitions)

*character-value*

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

Top

---

## Sequence number (SEQNBR)

Specifies the sequence numbers of the tape files used for the restore operation.

**Note:**When restoring document library objects (DLOs) from more than one auxiliary storage pool (ASP), a beginning and ending sequence number must be specified. The beginning and ending sequence numbers were shown in the completion message for the SAVDLO command used to create the tape media files. Or use the Display Tape (DSPTAP) command to find the sequence numbers of the QDOC and QDOCnnnn files, where nnnn is the ASP number.

### Single values

#### \*SEARCH

The tape is searched for the first data file with an identifier matching the **Label (LABEL)** parameter value and with contents of a minimum of one of the specified document library objects. If the last operation on the device specified ENDOPT(\*LEAVE) (that is, the tape is positioned at the location at which the last operation ended), the file search begins with the first data file beyond the current tape position. If ENDOPT(\*LEAVE) was not specified on the last operation (or if the tape has been rewound since the ENDOPT(\*LEAVE) operation), the search begins with the first data file on the volume.

#### Element 1: Beginning sequence number

1-16777215

Specify the sequence number of the first file used for the restore operation.

#### Element 2: Ending sequence number

#### \*ONLY

The ending sequence number is the same as the starting sequence number.

1-16777215

Specify the sequence number of the last file used for the restore operation.

Top

---

## End of media option (ENDOPT)

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

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

#### \*REWIND

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

#### \*LEAVE

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

#### \*UNLOAD

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

Top

---

## Label (LABEL)

Specifies the file label used to find the file that was written onto the media during the save operation.

**\*GEN** The system generates the default name of the file label for which to search.

*character-value*

Specify the file label of the file that contains the document library objects to be restored. A maximum of 17 alphanumeric characters can be used.

Top

---

## Save file (SAVF)

Specifies the save file used to restore the data.

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

### Qualifier 1: Save file

*name* Specify the name of save file to be used.

### Qualifier 2: Library

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

**\*CURLIB**

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

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

Top

---

## Optical file (OPTFILE)

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

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

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

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

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

*'optical-file-path-name'*

Specify the path name of the optical file.

Top

---

## Object name generation (NEWOBJ)

Specifies whether a new library-assigned name and system object name are generated for the folders and documents that are restored.

**\*SAME**

The library-assigned name and system object name do not change.

**\*NEW** A new library-assigned name and system object name are generated for each document or folder that is restored.

Top

---

## Date when saved (SAVDATE)

Specifies the date on which the document library objects were saved. If more than one version of the document library objects exist on the media, use this parameter to identify which version of the document library objects to restore. The date must be specified in the job date format. If separators are used, the value must be enclosed in apostrophes. If the SAVDATE parameter is not specified, the version of the documents and folders to be restored will be the first version found on the volume or the version found with the specified file label.

Top

---

## Time when saved (SAVTIME)

Specifies the time when the document library objects were saved. If more than one version of the document library objects exist on the media with the same value for the date saved, use this parameter to identify which version of the document library objects to restore. Specify the time as a 6-digit value, in the format hours, minutes, seconds (hhmmss). If separators are used, the value must be enclosed in apostrophes ('hh:mm:ss'). If a volume identifier is specified, but the **Time when saved (SAVTIME)** parameter is not specified, the version of the document library objects to be restored will be the first version found on the volume or the first version found with the specified file label.

This parameter is valid only if the **Date when saved (SAVDATE)** parameter is specified also.

Top

---

## Allow object differences (ALWOBJDIF)

Specifies whether the following differences encountered during a restore operation are allowed.

- Ownership—the owner of the object on the system is different than the owner of the object from the save operation.
- System object name—the system object name on the system does not match the system object name on the media.
- Authorization list linking—the object is being restored to a system different from the one on which it was saved.

The ALWOBJDIF parameter can be used to allow an object to be restored whose owner or object name on the system is different than on the media used for the restore operation. By specifying the **\*ALL** special value, an object with a different name is restored to the name on the media, while an object with a different owner keeps the owner name from the system instead of the media.

**Note:** To use this parameter, you need all object (**\*ALLOBJ**) authority.

### **\*NONE**

None of the differences described above are allowed on the restore operation. For authorization list cases, the object is restored, but the object is not linked to the authorization list, and public authority is set to **\*EXCLUDE**. For other cases, a diagnostic message is sent for the object, and the object is not restored.

**\*ALL** All of the differences described above are allowed for the restore operation. An informational message is sent, and the object is restored.



**Notes:**

- If the owners of the object do not match, the object is restored, but it will keep the ownership and authorities of the object on the system before the restore operation.
- If \*ALL is specified on this parameter, \*NEW cannot be specified for the **Object name generation (NEWOBJ)** parameter.
- If you are restoring objects to a system different from the one on which they were saved and the objects are secured by an authorization list, specifying \*ALL automatically links the objects to the authorization list. If the authorization list does not exist on the new system, a message that includes the name of the missing list is issued.

Top

---

## Saved from ASP number (SAVASP)

Specifies the number of the auxiliary storage pool (ASP) on media from which saved documents and folders are to be restored.

\*ANY The documents and folders saved in any ASP are restored.

**Note:**When restoring document library objects (DLOs) from more than one ASP, the sequence number (SEQNBR) parameter must be specified.

1-32 Specify the number of the ASP from which documents and folders are restored.

Top

---

## Restore to ASP number (RSTASP)

Specifies the number of the auxiliary storage pool (ASP) on media in which restored documents and folders are to be placed.

\*SAVASP

The documents and folders are placed in the same ASP from which they were saved.

1-32 Specify the number of the ASP in which restored documents and folders are placed.

Top

---

## Output (OUTPUT)

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

\*NONE

No output is created.

\*PRINT

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

\*OUTFILE

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

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

---

## File to receive output (OUTFILE)

Specifies the qualified name of the database file to which the information about the object is directed when \*OUTFILE is specified for the **Output (OUTPUT)** parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QAOJRSTO in QSYS with the format name QOJRST as a model.

### Qualifier 1: File to receive output

*name* Specify the name of the database file to which output from the command is directed. If this file does not exist, it is created in the specified library.

### Qualifier 2: Library

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

#### **\*CURLIB**

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

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

---

## Output member options (OUTMBR)

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

### Element 1: Member to receive output

#### **\*FIRST**

The first member in the file receives the output. If OUTMBR(\*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

*name* Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

### Element 2: Replace or add records

#### **\*REPLACE**

The existing records in the specified database file member are replaced by the new records.

**\*ADD** The new records are added to the existing information in the specified database file member.

---

## Examples

### Example 1: Restoring Documents with System Object Names

```
RSTDLO DLO(*SYSOBJNAM) DEV(TAP01) SYSOBJNAM(HZ83B55219)
```

This command restores the document named HZ83B55219 from the tape unit TAP01.

### Example 2: Restoring Documents from a Save Folder

```
RSTDLO DLO(A) DEV(TAP01) SAVFLR(X)
```

This command restores the document named A from folder X.

### Example 3: Restoring All Documents

```
RSTDLO DLO(*ALL) DEV(TAP01)
```

This command restores all documents and folders that are on the first tape file on tape unit TAP01.

### Example 4: Restoring a Folder Saved from the System ASP to a User ASP

```
RSTDLO DLO(*ALL) FLR(Y) SAVASP(1) RSTASP(2)
```

This command restores folder Y, which was saved from ASP 1, to user ASP 2. Folder Y must be deleted from ASP 1 before it can be restored to ASP 2.

### Example 5: Creating New Library-Assigned Name

```
RSTDLO DLO(*SYSOBJNAM) DEV(TAP01) SYSOBJNAM(HZ83B55219)  
NEWOBJ(*NEW)
```

This command restores document HZ83B55219 from tape unit TAP01 and gives it a new library-assigned name and a new system object name.

### Example 6: Renaming Documents

```
RSTDLO DLO(A B) DEV(TAP01) SAVFLR(C)  
RENAME(Y Z) RSTFLR(X)
```

This command restores documents A and B from within folder C. Document A is renamed to Y and document B is renamed to Z. The command then puts them in folder X.

### Example 7: Specifying Sequence Numbers

```
RSTDLO DLO(*ALL) DEV(tape-device-name) SAVFLR(A)  
SEQNBR(1 3) LABEL(*GEN)
```

This command restores all of folder A from tape files with the sequence numbers 1, 2, and 3, and the label QDOC or QDOCxxxx.

### Example 8: Specifying Allowed Differences

```
RSTDLO DLO(A) DEV(TAP01) SAVFLR(X) ALWBJDIF(*ALL)
```

This command restores document A from folder X. If document A in folder X exists on the system and the owner of the document on the system does not match the owner of the document being restored, the document is restored and the owner of the document on the system remains unchanged.

### Example 9: Reporting Information about Objects Restored and Not Restored

```
RSTDLO DLO(*ALL) DEV(TAP01) OUTPUT(*OUTFILE)  
OUTFILE(INF092) OUTMBR(FOURQT *ADD)
```

This command restores all documents and folders from the tape device TAP01. A list reporting information about objects restored and objects not restored is directed to the output file INFO92. The output is received in the member FOURQT as an addition to existing information in the member.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF370C

Not authorized to ALWOBJDIF parameter.

#### CPF3718

Restore command not valid for file &1.

#### CPF3728

Device &1 specified with other devices.

#### CPF3767

Device &1 not found.

#### CPF3780

Specified file for library &1 not found.

#### CPF3782

File &1 in &2 not a save file.

#### CPF381B

No DLOs restored to ASP &1.

#### CPF3812

Save file &1 in &2 in use.

#### CPF384D

Save or restore operation not allowed on ASP &1.

#### CPF8AB5

ASP &5 is not configured.

#### CPF8A47

Internal system objects in use.

#### CPF90AF

RSTFLR value not allowed.

#### CPF90A4

RENAME value not allowed.

#### CPF90B4

&1 folders restored to system, &2 not restored.

#### CPF90CD

Not authorized to restore distributions.

#### CPF90CF

Search index data base is damaged.

#### CPF90E0

Not enough authority for ALWOBJDIF(\*ALL).

#### CPF90E7

Document library objects not restored.

**CPF9003**  
&1 document library objects restored. &10 not restored.

**CPF905C**  
Error occurred trying to find a translation table.

**CPF9050**  
Ending sequence number not valid.

**CPF9069**  
User not permitted to restore into folder &1.

**CPF908A**  
Requester &1 not enrolled.

**CPF909B**  
&1 document library objects restored. &10 not restored.

**CPF9412**  
List of folder names not allowed with DLO parameter.

**CPF9810**  
Library &1 not found.

**CPF9812**  
File &1 in library &2 not found.

**CPF9820**  
Not authorized to use library &1.

**CPF9822**  
Not authorized to file &1 in library &2.

**CPF9825**  
Not authorized to device &1.

**CPF9830**  
Cannot assign library &1.

**CPF9831**  
Cannot assign device &1.

**CPF9845**  
Error occurred while opening file &1.

**CPF9846**  
Error while processing file &1 in library &2.

**CPF9850**  
Override of printer file &1 not allowed.

**CPF9851**  
Overflow value for file &1 in &2 too small.

**CPF9860**  
Error occurred during output file processing.

**CPF9899**  
Error occurred during processing of command.

**OPT1498**  
Volume name list exhausted on device &1.

**OPT1502**  
Attempted to process past the end of a multi-volume set.

**OPT1605**

Media or device error occurred.

[Top](#)

---

## Restore Library (RSTLIB)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Restore Library (RSTLIB) command restores to the system one library or a group of libraries that was saved by the Save Library (SAVLIB) command. The Restore Library (RSTLIB) command restores the whole library, which includes the library description, object descriptions, and contents of the objects in the library.

For job queues, message queues, user queues, and logical files, only the object descriptions are restored, because only the definitions are saved. Also, logical file access paths may be restored if they were saved. More information on restoring access paths is in the Database information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>. The contents of spooled files on output queues can be restored, if they were saved, by specifying \*NEW for the **Spooled file data (SPLFDTA)** parameter.

This command can be used to restore libraries whose objects had their storage freed by the corresponding SAVLIB command of the restore operation, or libraries deleted by the Delete Library (DLTLIB) command. If the data portions of the objects in the saved libraries were not freed, each library is copied into the same area of storage that it previously occupied. If the storage was freed, the system finds the needed storage to store the library contents (the object description and data portion of every file, module, program, service program, Structured Query Language (SQL) package, and journal receiver in the library). If the library does not exist on the system because it has been deleted or is being restored on a different system, the system must find the storage for everything that is in the library, including the library description.

When the owner profile does not exist on the system, the user profile of the system default owner (QDFTOWN) becomes the default owner of any object being restored in the system.

If an object already exists in the library in which it is being restored, the public and private authorities of the existing object are retained. If the object does not exist in the library, all public authorities are restored, but private authorities must be granted again. For an existing output queue object that is actively spooling during the restore operation, or a data queue that already exists in the library, the object is not restored, and a diagnostic message is sent.

If an object is being restored over an existing object on the system, the object auditing value of the existing object is kept. If the object is being restored as new to the system, the object auditing value is restored from the media. Additionally, if the object is a library, the default auditing value for each object created in the library is restored if the library is being restored as new; otherwise, the default auditing value is restored from the media.

### Notes:

1. To restore a save file to a library where it does not already exist, you must have \*USE authority to the Create Save File (CRTSAVF) command.
2. The RSTLIB command ignores all file overrides currently in effect for the job, except the overrides for the restore output file.

### Restrictions:

- This command is shipped with no public authority (\*EXCLUDE).
- You must have save system (\*SAVSYS) special authority, or have all of the following object authorities:

1. Read (\*READ) and add (\*ADD) authorities for, or be the owner of, each library specified.
  2. Object existence (\*OBJEXIST) authority for, or be the owner of, each object in the library if the object already exists in the library on the system. \*OBJEXIST and use (\*USE) authorities are required for message queue objects. If the object does not exist, add (\*ADD) authority for the user profiles that own the objects to be restored. For spooled file data, \*OBJEXIST authority is required for the output queue to which it is restored.  
If you do not have the correct authority for all the libraries and objects specified, only those for which you have authority are restored.
  3. If VOL(\*SAVVOL) is specified, \*USE authority to the saved-from library.
  4. \*USE authority for the save file is required when restoring libraries from a save file. \*USE authority for the device description and the device file are required when restoring libraries from a tape or optical device.
- When using a media definition, you must have \*USE authority to the media definition and execute (\*EXECUTE) authority to the media definition library.
  - When saving or restoring to an existing database file using the OUTFILE parameter, you must have \*EXECUTE authority to the library where the output file is located.
  - The current versions of programs on the system should not be run while the library that contains those programs is being restored. If any OPM program is running while it is being restored, the running program may fail or behave unpredictably.

Top

## Parameters

Keyword	Description	Choices	Notes
SAVLIB	Saved library	Single values: *NONSYS, *ALLUSR, *IBM, *ANY Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
DEV	Device	Single values: *SAVE, *MEDDFN Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 2
VOL	Volume identifier	Single values: *MOUNTED, *SAVVOL Other values (up to 75 repetitions): <i>Character value</i>	Optional, Positional 3
SEQNBR	Sequence number	1-16777215, *SEARCH	Optional, Positional 4
LABEL	Label	<i>Character value</i> , *SAVLIB	Optional
ENDOPT	End of media option	*REWIND, *LEAVE, *UNLOAD	Optional
STRLIB	Starting library	<i>Name</i> , *FIRST	Optional
SAVF	Save file	<i>Qualified object name</i>	Optional
	Qualifier 1: Save file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
MEDDFN	Media definition	<i>Qualified object name</i>	Optional
	Qualifier 1: Media definition	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
OPTFILE	Optical file	<i>Path name</i> , _	Optional
OMITLIB	Libraries to omit	Single values: *NONE Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional
OPTION	Option	*ALL, *NEW, *OLD, *FREE	Optional
MBROPT	Data base member option	*MATCH, *ALL, *NEW, *OLD	Optional
SPLFDTA	Spooled file data	*NEW, *NONE	Optional
SAVDATE	Date when saved	<i>Date</i>	Optional
SAVTIME	Time when saved	<i>Time</i>	Optional



Keyword	Description	Choices	Notes
ALWOBJDIF	Allow object differences	Single values: *NONE, *ALL Other values (up to 4 repetitions): *AUTL, *FILELVL, *OWNER, *PGP	Optional
FRCOBJCVN	Force object conversion	Single values: *SYSVAL, *NO Other values: <i>Element list</i>	Optional
	Element 1: Convert during restore	*YES	
	Element 2: Objects to convert	*RQD, *ALL	
RSTLIB	Restore to library	<i>Name</i> , *SAVLIB	Optional
RSTASPDEV	Restore to ASP device	<i>Name</i> , *SAVASPDEV	Optional
RSTASP	Restore to ASP number	1-32, *SAVASP	Optional
OUTPUT	Output	*NONE, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
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	
INFTYPE	Type of output information	*OBJ, *MBR	Optional
OMITOBJ	Objects to omit	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Generic name, name</i> , *NONE, *ALL	
	Qualifier 2: Library	<i>Generic name, name</i> , *ALL	
	Element 2: Object type	<i>Character value</i> , *ALL	

Top

## Saved library (SAVLIB)

Specifies the libraries to be restored to the system.

### Notes:

- If you specify more than one library, or a special value, or a generic value:
  - The **Device (DEV)** parameter must not specify \*SAVE.
  - The **Restore to library (RSTLIB)** parameter must specify \*SAVLIB.
  - The **Label (LABEL)** parameter must specify \*SAVLIB.
  - The **Optical file (OPTFILE)** parameter must specify \* or end with /\*.
- For values \*NONSYS and \*IBM, all other operations on the system must be ended before this option is specified. This requires ending all subsystems through the End Subsystem (ENDSBS) command or the End System (ENDSYS) command.
- The tape or optical file that you select to restore may contain a library that does not match the value that you specify for this parameter. This parameter is used to determine the default LABEL value for a tape file and the default OPTFILE value for an optical file, but the restore operation does not verify that this parameter matches the saved library.

This is a required parameter.

## Single values

### \*NONSYS

Libraries saved by the Save Library (SAVLIB) command with LIB(\*NONSYS) specified are restored.

You can do a RSTLIB SAVLIB(\*IBM) and a RSTLIB SAVLIB(\*ALLUSR) from a SAVLIB LIB(\*NONSYS).

### \*ALLUSR

All user libraries are restored. All libraries with names that do not begin with the letter Q are restored except for the following:

#CGULIB	#DSULIB	#SEULIB
#COBLIB	#RPGLIB	
#DFULIB	#SDALIB	

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered "user libraries", and are also restored:

QDSNX	QRCLxxxxx	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	

### \*IBM Restores all IBM-supplied libraries except for the following:

QDOC	QRCYxxxxx	QTEMP	QUSRDRARS
QDOCxxxx	QRECOVERY	QUSER38	QUSRSYS
QDSNX	QRPLOBJ	QUSRADSM	QUSRVI
QGPL	QRPLxxxxx	QUSRBRM	QUSRVxRxMx
QGPL38	QSPL	QUSRDIRCL	
QMGTC	QSPLxxxx	QUSRDIRDB	
QMGTC2	QSRV	QUSRIJS	
QMPGDATA	QSRVAGT	QUSRINFSKR	
QMOMDATA	QSYS	QUSRNOTES	
QMOMPROC	QSYSxxxxx	QUSROND	
QPFRDATA	QSYS2	QUSRPOSGS	
QRCL	QSYS2xxxxx	QUSRPOSSA	
QRCLxxxxx	QS36F	QUSRPYMSVR	

**Note:** A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

The following libraries with names that do not begin with the letter Q are also restored:

#CGULIB	#DSULIB	#SEULIB
#COBLIB	#RPGLIB	
#DFULIB	#SDALIB	

### \*ANY Restores the first version of all saved libraries found on the tape beginning with the sequence number specified for the **Sequence number (SEQNBR)** parameter, or restores all saved libraries found on the optical media in the directory specified for the **Optical file (OPTFILE)** parameter.

## Other values (up to 300 repetitions)

### *generic-name*

Specify the generic name of the library to be restored.

**Note:** A generic name is specified as a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

*name* Specify the name of the library to be restored. The name of the library being restored must be the same as the name that was used when the library was saved.

Top

---

## Device (DEV)

Specifies the name of the device used for the restore operation. The device name must already be known on the system by a device description.

This is a required parameter.

### Single values

\***SAVF** The restore operation is done using the save file specified for the **Save file (SAVF)** parameter.

\***MEDDFN**

The restore operation is done using the devices and media identified in the media definition specified for the **Media definition (MEDDFN)** parameter.

### Other values

*optical-device-name*

Specify the name of the optical device used for the restore operation.

*tape-media-library-device-name*

Specify the name of the tape media library device used for the restore operation.

*tape-device-name*

Specify the names of one or more tape devices used for the restore operation. If a virtual tape device is used, it must be the only device specified. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order in which they are used. When more than one tape volume is to be restored, using more than one tape device permits one tape volume to be rewound while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Top

---

## Volume identifier (VOL)

Specifies the volume identifiers of the media or the cartridge identifiers of tapes in a tape media library device, from which the objects are being restored. The volumes must be in the same order as they were when the data was saved. The volume that contains the beginning of the file to be restored should be placed in the device.

**Note:** The first version of the saved library that is found on the media is restored, unless a specific version is identified by the **Date when saved (SAVDATE)** and **Time when saved (SAVTIME)** parameters or, for tape, the **Sequence number (SEQNBR)** parameter.

### Single values

\***MOUNTED**

The objects are restored from the volumes placed in the device specified for the **Device (DEV)**

parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

**Note:** This value cannot be specified when using an optical media library device.

#### **\*SAVVOL**

The system, by using the save or restore history information, determines which volumes contain the most recently saved version of the library. If the device type of the device specified for the **Device DEV** parameter does not match the device type of the most recently saved version of the library, an error message is sent to the user, and the function is ended. If \*SAVVOL is specified, the SAVDATE and SAVTIME parameters cannot be specified. If \*SAVVOL is specified, \*SEARCH must be specified for the SEQNBR parameter.

#### **Other values (up to 75 repetitions)**

##### *character-value*

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to restore the data.

Top

---

## **Sequence number (SEQNBR)**

Specifies, when tape is used, which sequence number is used for the restore operation.

#### **\*SEARCH**

The volume in the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the object is restored. If the last operation on the device specified \*LEAVE for the **End of tape option (ENDOPT)** parameter, indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If \*LEAVE was not used for the **End of tape option (ENDOPT)** parameter of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

#### **1-16777215**

Specify the sequence number of the file to be used for the restore operation.

If \*NONSYS, \*ALLUSR, or \*IBM is specified for the **Saved library (SAVLIB)** parameter, the sequence number specifies the location of the file QFILE. The QFILE file is at the beginning of the \*NONSYS, \*ALLUSR, or \*IBM save operation. The QFILE file contains the list of libraries saved.

Top

---

## **Label (LABEL)**

Specifies the name that identifies the data file to be used for the restore operation. This label must have been specified on the save command.

#### **\*SAVLIB**

The file label is the name specified for the **Saved library (SAVLIB)** parameter.

##### *character-value*

Specify the data file identifier of the data file used for the restore operation. A maximum of 17 characters can be used. This option is valid only for a single-library restore.

Top

---

## End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the restore operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

**Note:** This parameter is valid only if a tape or optical device name is specified for the DEV parameter. For optical devices, \*UNLOAD is the only special value supported, \*REWIND and \*LEAVE will be ignored.

### \*REWIND

The tape is automatically rewound, but not unloaded, after the operation has ended.

### \*LEAVE

The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

### \*UNLOAD

The tape is automatically rewound and unloaded after the operation ends. Some optical devices will eject the volume after the operation ends.

Top

---

## Starting library (STRLIB)

Specifies the name of the starting library for a \*NONSYS, \*IBM, or \*ALLUSR restore operation.

If an unrecoverable media error occurs during the restore operation, this parameter can be used to restart the operation.

**Note:** In the recovery steps that follow, \*NONSYS is specified for the **Saved library (SAVLIB)** parameter of this command. If you are restoring IBM-supplied libraries or all user-created libraries and IBM-supplied libraries, specify \*IBM or \*ALLUSR instead.

The basic recovery steps for a restore operation are:

1. Look at the job log to determine the library where the previous restore library (RSTLIB SAVLIB(\*NONSYS)) command failed. Find the last library restored, which is indicated by a successful restore completion message.
2. Load the first tape of the SAVLIB LIB(\*NONSYS) media.
3. Type the following command, and then press the Enter key:  
RSTLIB SAVLIB(\*NONSYS) DEV(TAP01) ENDOPT(\*LEAVE)  
STRLIB(library-name) OMITLIB(library-name)  
where the *library-name* for the STRLIB and OMITLIB parameters is the library where the last RSTLIB failed. This starts the restore operation on the library after the library where the restore operation failed.
4. When you are prompted, load the volume containing the starting library.
5. After the restore operation is complete, restore the library where the restore operation failed using the media from a previous save operation.

**Note:** Consider removing the tape with the media error from the next save rotation cycle to avoid another tape error.

**Note:** This parameter is valid only if \*NONSYS, \*IBM, or \*ALLUSR is specified for the SAVLIB parameter.

### \*FIRST

The restore operation begins with the first library saved.

*name* Specify the name of the library with which to start the restore operation.

Top

---

## Save file (SAVF)

Specifies the save file used to restore the data.

**Note:** A value must be specified for this parameter if \*SAVF is specified for the **Device (DEV)** parameter.

### Qualifier 1: Save file

*name* Specify the name of save file to be used.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

#### **\*CURLIB**

The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

*name* Specify the name of the library where the save file is located.

Top

---

## Media definition (MEDDFN)

Specifies the media definition (\*MEDDFN) object that identifies the devices and media used to restore the data. For information about creating and using a media definition, see the Backup and Recovery book, SC41-5304, and the Create Media Definition API in the System API Reference information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

If a media definition is specified, the VOL, SEQNBR, SAVF, and OPTFILE parameters cannot be specified. The volume identifiers and sequence numbers are specified in the media definition.

### Qualifier 1: Media definition

*name* Specify the name of the media definition to be used.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

#### **\*CURLIB**

The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched.

*name* Specify the name of the library to be searched.

Top

---

## Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume.

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.

**\* \_** The system searches the root directory of the optical volume for the default name generated by the corresponding save operation.

**'optical-directory-path-name\*'**

The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

**'optical-file-path-name'**

Specify the path name of the optical file.

Top

---

## Libraries to omit (OMITLIB)

Specifies a list of libraries to be excluded from the restore operation.

### Single values

**\*NONE**

No libraries are excluded from the restore operation.

### Other values (up to 300 repetitions)

**generic-name**

Specify the generic name of the libraries to be excluded.

**Note:** A generic name is specified as a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

**library-name**

Specify the name of the library to be excluded from the restore operation.

Top

---

## Option (OPTION)

Specifies how to handle restoring each object.

- \*ALL** All the objects in the saved library are restored to the library. Objects in the saved library replace the current versions in the system library. Objects not having a current version are added to the system library. Objects presently in the library, but not on the media, remain in the library.
- \*NEW** Only the objects in the saved library that do not exist in the current version of the system library are added to the library. Only objects not known to the system library are restored; known objects are not restored. This option restores objects that were deleted after they were saved or that are new to this library. If any saved objects have a version already in the system library, they are not restored, and an informational message is sent for each one, but the restore operation continues.
- \*OLD** Only the objects in the library having a saved version are restored; that is, the version of each object currently in the library is replaced by the saved version. Only objects known to the library are restored. If any saved objects are no longer part of the online version of the library, they are not added to the library; an informational message is sent for each one, but the restore continues.
- \*FREE** The saved objects are restored only if they exist in the system library with their space freed. The saved version of each object is restored on the system in its previously freed space. This option restores objects that had their space freed when they were saved. If any saved objects are no

longer part of the current version of the library, or if the space is not free for any object, the object is not restored and an informational message is sent for each one. The restore operation continues, and all of the freed objects are restored.

Top

---

## Data base member option (MBROPT)

Specifies, for database files that exist on the system, which members are restored. If \*MATCH is used, the member list in the saved file must match, member for member, the current version on the system. All members are restored for files that do not exist, if the file is restored.

### \*MATCH

The saved members are restored if the lists of the members where they exist match, member for member, the lists of the current system version. MBROPT(\*MATCH) is not valid when \*ALL is specified for the **Allow object differences (ALWOBJDIF)** parameter.

\*ALL All members in the saved file are restored.

\*NEW Only new members (members not known to the system) are restored.

\*OLD Only members already known to the system are restored.

Top

---

## Spooled file data (SPLFDTA)

Specifies whether to restore spooled file data and attributes.

\*NEW For each output queue that is restored, spooled file data that was saved with the output queue is restored, if it does not already exist on the system.

### \*NONE

No spooled file data is restored.

Top

---

## Date when saved (SAVDATE)

Specifies the date when the library was saved. If the most recently saved version is not restored, or if more than one saved version is on the volumes, type the date that specifies which version of the library is restored.

If a volume identifier or \*MOUNTED is specified for the **Volume identifier (VOL)** parameter, but this parameter is not specified, the first version of the library found on the volume is restored. This parameter is not valid with VOL(\*SAVVOL).

*date* Specify the date that the library to be restored was saved. The date must be entered in the job date format.

Top

---

## Time when saved (SAVTIME)

Specifies the time when the library was saved, if the current version is not restored.



If a volume identifier or \*MOUNTED is specified for the **Volume identifier (VOL)** parameter, but this parameter is not, the first version of the library found on the volume is restored. This parameter is valid only if the **Date when saved (SAVDATE)** parameter is also specified.

**time** Specify the time that the library to be restored was saved. The time can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Top

---

## Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

### Notes:

1. You must have all object (\*ALLOBJ) special authority to specify any value other than \*NONE for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- **Authorization list:** The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.  
**Note:** This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.
- **File level id:** The creation date and time of the database file on the system does not match the creation date and time of the file that was saved.
- **Member level id:** The creation date and time of the database file member on the system does not match the creation date and time of the member that was saved.
- **Ownership:** The owner of an object on the system is different than the owner of an object from the save operation.
- **Primary Group:** The primary group of an object on the system is different than the primary group of an object from the save operation.

### Single values

#### \*NONE

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

- **\*ALL** All of the differences listed above are allowed on the restore operation. File level id and member level id differences are handled differently than the \*FILELVL value. If there is a file level difference and \*ALL is specified for the **Data base member option (MBROPT)** parameter, the existing version of the file is renamed and the saved version of the file is restored. If there is a member level difference, the existing version of the member is renamed and the saved version of the member is restored. This value will restore the saved data, but the result may not be correct. You will need to choose whether the restored data or the renamed data is correct, and you will

need to make the necessary corrections to the database. For other differences, see the description of each individual value to determine how differences are handled.

### Other values (up to 4 repetitions)

#### \*AUTL

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to \*EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to \*EXCLUDE.

#### \*FILELVL

File level id and member level id differences are allowed. An attempt will be made to restore existing physical files even though the physical file on the save media may have a different file level id or member level id than the physical file on the system. The physical file data will only be restored for those physical files whose format level identifiers on the save media match the format level identifiers of the corresponding physical file on the system.

If this value is not specified, file level id and member level id differences are not allowed. If an object already exists on the system with a different file level id or member level id than the saved object, the object is not restored.

#### \*OWNER

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

\*PGP Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Top

---

## Force object conversion (FRCOBJCVN)

Specifies whether to convert user objects to the format required for use in the current version of the operating system when the objects are restored.

### Notes:

1. This parameter applies only to user objects of the \*MODULE, \*PGM, \*SRVPGM, and \*SQLPKG object types.
2. An object must have creation data (either observable or unobservable) to be converted.
3. If an object needs to be converted (because it is formatted for an earlier version of the operating system), but is not converted during this restore operation, the object is automatically converted the first time it is used.

## Single values

### \*SYSVAL

The objects are converted based on the value of the QFRCCVNRST system value.

**\*NO** The objects are not converted during the restore operation.

**Note:** If FRCOBJCVN(\*NO) is specified, then the QFRCCVNRST system value must have a value of either "0" or "1".

## Element 1: Convert during restore

**\*YES** The objects are converted during the restore operation.

### Notes:

1. If FRCOBJCVN(\*YES \*RQD) is specified, then the QFRCCVNRST system value must have a value of "0", "1", or "2". FRCOBJCVN(\*YES \*RQD) will override a QFRCCVNRST value of "0" or "1". If FRCOBJCVN(\*YES \*ALL) is specified, then QFRCCVNRST can have any valid value and FRCOBJCVN(\*YES \*ALL) overrides the QFRCCVNRST system value.
2. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

## Element 2: Objects to convert

**\*RQD** The objects are converted only if they require conversion to be used by the current operating system. If the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

**\*ALL** All objects are converted regardless of their current format, including objects already in the current format. However, if the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

Top

---

## Restore to library (RSTLIB)

Specifies whether the library contents are restored to the same library in which they were saved, or to a different library.

### \*SAVLIB

The library contents are restored to the same library or libraries in which they were saved.

**name** Specify the name of the library where the saved library contents are restored. If \*NONSYS, \*IBM, or \*ALLUSR is specified for the **Saved library (SAVLIB)** parameter, a library name cannot be specified for this parameter.

**Note:** If a Structured Query Language (SQL) database is restored to a library other than the one in which it was saved, the journals are not restored.

Top

---

## Restore to ASP device (RSTASPDEV)

Specifies the auxiliary storage pool (ASP) device to which the data is to be restored.

**Note:** You can specify either the RSTASPDEV parameter or the RSTASP parameter, but not both.

### \*SAVASPDEV

The data is restored to the same ASP from which it was saved.

*name* Specify the name of the ASP device to be used.

Top

---

## Restore to ASP number (RSTASP)

Specifies whether objects are restored to the auxiliary storage pool (ASP) from which they were saved or to the system ASP (ASP number 1) or to a basic user ASP (ASP numbers 2 through 32).

Some objects cannot be restored to user ASPs. More information about object types which can be restored to user ASPs is in the Backup and Recovery book, SC41-5304. If the library exists in, or is being restored to the system ASP, journals, journal receivers, and save files can be restored to basic user ASPs. All other object types will be restored to the ASP of the library.

**ATTENTION: System or product libraries (libraries that begin with a Q or #) must not be created in or restored to a user ASP. Doing so can cause unpredictable results.**

### \*SAVASP

The objects are restored to the ASP from which they were saved.

**1-32** Specifies the ASP number. When the specified ASP is 1, the specified objects are restored to the system ASP, and when the specified ASP is 2 through 32, the objects are restored to the basic user ASP specified.

Top

---

## Output (OUTPUT)

Specifies whether a listing that shows information about the status of the objects is created and directed to an output file. The listing shows the restore information and shows all objects restored, not restored, and excluded. Information about each object's security is listed for the restored objects.

### \*NONE

No output is created.

### \*PRINT

The output is printed with the job's spooled output.

### \*OUTFILE

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

**Note:** You must specify a database file name for the OUTFILE parameter when OUTPUT(\*OUTFILE) is specified.

Top

---

## File to receive output (OUTFILE)

Specifies the database file to which the information is directed when \*OUTFILE is specified for the **Output (OUTPUT)** parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QASRRSTO in QSYS with the format name QSRRST as a model.

### Qualifier 1: File to receive output

*name* Specify the name of the database file to which output from the command is directed. If this file does not exist, it is created in the specified library.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the thread is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

*name* Specify the name of the library to be searched.

Top

---

## Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when \*OUTFILE is specified for the **Output (OUTPUT)** parameter.

### Element 1: Member to receive output

**\*FIRST**

The first member in the file receives the output. If OUTMBR(\*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

*name* Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

### Element 2: Replace or add records

**\*REPLACE**

The existing records in the specified database file member are replaced by the new records.

**\*ADD** The new records are added to the existing information in the specified database file member.

Top

---

## Type of output information (INFTYPE)

Specifies the type of information which is printed or directed to the database file.

**\*OBJ** The list contains an entry for each object requested to be restored.

**\*MBR** The list contains an entry for each object, database file member, and spooled file requested to be restored.

Top

---

## Objects to omit (OMITOBJ)

Specifies the objects to be excluded from the operation. Up to 300 objects or generic object values can be specified.

### Element 1: Object

#### Qualifier 1: Object

##### \*NONE

No objects are excluded from the operation.

**\*ALL** All objects of the specified object type are excluded from the operation.

##### *generic-name*

Specify the generic name of the objects to be excluded.

**Note:** A generic name is specified as a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

*name* Specify the name of the object to be excluded from the operation.

#### Qualifier 2: Library

\*ALL The specified objects are excluded from all libraries that are part of the operation.

##### *generic-name*

Specify the generic name of the libraries that contain objects to be excluded.

**Note:** A generic name is specified as a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

*name* Specify the name of the library that contains the object to be excluded from the operation.

### Element 2: Object type

\*ALL All object types are excluded from the operation, depending on the value specified for the object name.

##### *character-value*

Specify the object type of the objects to be excluded from the operation.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Top

---

## Examples

### Example 1: Restoring New Objects

```
RSTLIB SAVLIB(JOE) DEV(TAP01) OPTION(*NEW)
```

This command restores the saved version of library JOE from tape device TAP01. The only objects that are restored in the library are new objects (ones that were in the library when they were saved and later deleted).

### Example 2: Printing Output

```
RSTLIB SAVLIB(*NONSYS) DEV(TAP01) OUTPUT(*PRINT)
```

This command restores all the saved nonsystem libraries to the system from tape. The contents of the libraries are restored exactly as they were saved. New objects (on tape) are added to the system; old objects in the system are overlaid by the version of the old objects on tape. Because OUTPUT(\*PRINT) is specified, a printout of all objects (restored and not restored) for each library, is sent to the printer with the job's spooled output. Each library after the first library starts on a new page. After each library, a completion message states how many objects were restored and how many were not restored. At the end of a list, a final completion message states how many libraries were restored and how many were not restored.

### Example 3: Specifying Where the Restore Operation Begins

```
RSTLIB SAVLIB(*NONSYS) DEV(TAP01) STRLIB(MIKESLIB)
```

This command restores the saved nonsystem libraries beginning with library MIKESLIB from the tape device, TAP01. The *first* tape of the nonsystem save must be loaded. An inquiry message instructs the user to load the tape containing MIKESLIB. If necessary, the same message is sent until the tape containing MIKESLIB is found.

### Example 4: Restoring a Version From a Specific Date and Time

```
RSTLIB SAVLIB(PAYROLL) DEV(TAP01) SAVDATE(060193)  
SAVTIME(103214) RSTLIB(OLDPAY) VOL(PAY)
```

This command restores the version of the PAYROLL library from the device TAP01, whose volume identifier is PAY. The version to be restored was saved at 10:32:14 on the date 06/01/93. All of the objects in the saved PAYROLL library are restored to the library OLDPAY. All new files are restored. Old files are restored only if the member lists of the files on the tape match the member lists of the files on the system.

### Example 5: Restoring From Multiple Tape Volumes

```
RSTLIB SAVLIB(QGPL) DEV(TAP01) VOL(QGPL QGPL)
```

This command restores the QGPL library from two tape volumes both named QGPL. Even though the volume identifiers are the same, they must both be specified.

### Example 6: Restoring From Multiple Tape Devices Serially

```
RSTLIB SAVLIB(USRLIB) DEV(TAP01 TAP02 TAP03)  
VOL(USRA USRB USRC USRD) ENDOPT(*UNLOAD)
```

This command restores library USRLIB from four volumes on three tape devices. Volume USRA is put on tape device TAP01, volume USRB on TAP02, volume USRC on TAP03, and volume USRD on TAP01. The operator removes volume USRA from TAP01, so that TAP01 can be used by volume USRD. If the tape volumes are put in the wrong order, an error message is sent to the system operator message queue.

### Example 7: Restoring From Multiple Devices in Parallel

```
RSTLIB SAVLIB(USRLIB) DEV(*MEDDFN) MEDDFN(LIBA/MEDDFNA)
```

This command restores library USRLIB using the devices and volumes specified in the media definition named MEDDFNA in library LIBA. For information about creating and using a media definition, see the **Backup, Recovery, and Availability** topic in the Information Center.



### Example 8: Restoring a Specific Version

```
RSTLIB SAVLIB(LIB1) DEV(TAP01) MBROPT(*ALL)
SAVDATE(082392) SAVTIME(123251)
RSTLIB(LIB2) OUTPUT(*PRINT)
```

This command restores the version of library LIB1 from the device TAP01. The version to be restored was saved at 12:32:51 on the date 08/23/92. All of the objects in the saved library LIB1 are restored to library LIB2. A list of restored objects and those not restored is given. All files and file members are restored.

### Example 9: Restoring a Library From a Save File

```
RSTLIB SAVLIB(LIB1) DEV(*SAVF) SAVF(SAVF1)
```

This command restores library LIB1 from the save file SAVF1.

### Example 10: Restoring to a Basic User ASP

```
RSTLIB SAVLIB(LIB1) DEV(*SAVF) SAVF(SAVF1) RSTASP(2)
```

This command restores the library named LIB1 from the save file named SAVF1. The library and all objects in the saved version of LIB1 are restored to basic user auxiliary storage pool (ASP) 2 unless:

- The library already exists in a different ASP.
- ASP 2 contains a journal, journal receiver, or SAVF which is part of a library in the system ASP.
- ASP 2 does not exist on the system.
- There are object types in the library which cannot be restored to user ASPs. These objects will not be restored.

### Example 11: Restoring to an Independent ASP

```
RSTLIB SAVLIB(LIB1) DEV(*SAVF) SAVF(SAVF1)
RSTASPDEV(DIVISION1)
```

This command restores the library named LIB1 from the save file named SAVF1. The library and all objects in the saved version of LIB1 are restored to the independent ASP named DIVISION1 unless:

- The library already exists in the system ASP, a basic user ASP, or a different independent ASP in the same linked ASP group as DIVISION1.
- ASP DIVISION1 is not available on the system.
- There are object types in the library which cannot be restored to independent ASPs. These objects will not be restored.

### Example 12: Restoring More Than One Library From Tape

```
RSTLIB SAVLIB(*ANY) DEV(TAP01) SEQNBR(75)
```

This command restores the first saved version of each library from tape device TAP01 starting at tape file sequence number 75.

### Example 13: Restoring More Than One Library From Optical

```
RSTLIB SAVLIB(A* B*) DEV(OPT01) OPTFILE('/BACKUP/*')
```

This command restores all libraries whose names start with A or B that are found in directory BACKUP on the optical volume in device OPT01.

### Example 14: Restoring Spooled Files

```
RSTLIB SAVLIB(JOE) DEV(TAP01) SPLFDTA(*NEW)
```

This command restores the saved version of library JOE from tape device TAP01. Any spooled files that do not currently exist on the system and that were saved with output queues in this library are restored.



---

## Error messages

### \*ESCAPE Messages

**CPF370C**

Not authorized to ALWOBJDIF parameter.

**CPF3705**

&2 &1 in &3 not journaled.

**CPF3706**

&2 &1 not restored to library &3.

**CPF3707**

Save file &1 in &2 contains no data.

**CPF3709**

Tape devices do not support same densities.

**CPF372C**

Library &1 not restored to ASP &2.

**CPF3727**

Duplicate device &1 specified on device name list.

**CPF3728**

Device &1 specified with other devices.

**CPF373E**

Library &1 not restored to ASP &2.

**CPF3730**

Not authorized to &2 &1 in library &3.

**CPF3731**

Cannot use &2 &1 in library &3.

**CPF3732**

&2 &1 status error during restore operation.

**CPF3733**

&2 &1 in &3 previously damaged.

**CPF3738**

Device &1 used for save or restore is damaged.

**CPF3739**

Database file &1 member in &3 damaged.

**CPF3740**

Object &1 type &2 in &3 not found.

**CPF3743**

File cannot be restored, displayed, or listed.

**CPF3748**

Object information for library &1 damaged.

**CPF375F**

File not selected. Cannot restore from save type file &1.

**CPF3752**

No record of save operation exists for library &1.

**CPF3757**  
&2 &1 not restored to &3.

**CPF3758**  
&2 &1 not restored to &3.

**CPF376B**  
File &1 not found.

**CPF3767**  
Device &1 not found.

**CPF3768**  
Device &1 not valid for command.

**CPF3769**  
File found on media not save/restore file.

**CPF3770**  
No objects saved or restored for library &1.

**CPF3773**  
&1 objects restored. &2 not restored to &4.

**CPF3779**  
&1 libraries restored; &4 partially restored; &2 not restored.

**CPF378B**  
Library &1 not created.

**CPF3780**  
Specified file for library &1 not found.

**CPF3782**  
File &1 in &2 not a save file.

**CPF3783**  
Cannot determine VOL(\*SAVVOL) location. No objects restored.

**CPF3784**  
Restore device specified in the DEV parameter does not match VOL(\*SAVVOL) device.

**CPF3785**  
Not all subsystems ended.

**CPF3791**  
While processing &2 &1 in &3, encountered end of file &4.

**CPF3793**  
Machine or ASP storage limit reached.

**CPF3794**  
Save or restore operation ended unsuccessfully.

**CPF3796**  
Storage limit exceeded for user profile &4.

**CPF37A5**  
RSTASP &1 is not online.

**CPF380E**  
User profile &1 not restored. Not able to create UID or GID.

**CPF3805**  
Objects from save file &1 in &2 not restored.

**CPF3807**  
Data decompression error for save file &1 in &2.

**CPF3812**  
Save file &1 in &2 in use.

**CPF3818**  
Starting library &1 not found.

**CPF384F**  
&2 &1 not restored to library &3.

**CPF3894**  
Cancel reply received for message &1.

**CPF38A2**  
ASP device &1 not correct.

**CPF38A6**  
Only one library allowed with specified parameters.

**CPF5729**  
Not able to allocate object &1.

**CPF9809**  
Library &1 cannot be accessed.

**CPF9812**  
File &1 in library &2 not found.

**CPF9814**  
Device &1 not found.

**CPF9820**  
Not authorized to use library &1.

**CPF9822**  
Not authorized to file &1 in library &2.

**CPF9825**  
Not authorized to device &1.

**CPF9829**  
Auxiliary storage pool &1 not found.

**CPFB8ED**  
Device description &1 not correct for operation.

**OPT1498**  
Volume name list exhausted on device &1.

**OPT1502**  
Attempted to process past the end of a multi-volume set.

**OPT1605**  
Media or device error occurred.

**\*STATUS Messages**

**CPF3770**  
No objects saved or restored for library &1.

**CPF3773**  
&1 objects restored. &2 not restored to &4.



---

## Restore Licensed Program (RSTLICPGM)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Restore Licensed Program (RSTLICPGM) command loads or restores a licensed program for initial installation, for new-release installation, or for recovery.

### Restrictions:

1. This command is shipped with public \*EXCLUDE authority.
2. To use this command, the user must have \*SECADM and \*ALLOBJ authority.
3. If this command is used to restore a program in the licensed program, the copy of the program currently in the system should not be running while the program is being restored. If this occurs, the processing program is ended abnormally.
4. If other objects of the licensed program are in use, they are not restored.
5. With the exception of overrides for the restore operation printing OUTPUT(\*PRINT), this command ignores all file overrides currently in effect for the job.
6. Some licensed programs are restored only if the user is enrolled in the system distribution directory. See the publications for each licensed program for a description of this restriction.
7. This command does not restore code and language objects for the base operating system.
8. This command does not support the use of user ASPs (auxiliary storage pools). All objects supplied by a licensed program must remain in the system ASP.
9. At most one optical device, one save file, one virtual tape device or one tape media library device can be specified.

Top

---

## Parameters

Keyword	Description	Choices	Notes
LICPGM	Product	<i>Character value</i>	Required, Positional 1
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 2
OPTION	Optional part to be restored	* <u>BASE</u> , 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 01, 02, 03, 04, 05, 06, 07, 08, 09	Optional
RSTOBJ	Type of object to be restored	* <u>ALL</u> , *PGM, *LNG	Optional
LNG	Language for licensed program	<i>Character value</i> , * <u>PRIMARY</u> , *SAVVOL	Optional
OUTPUT	Output	* <u>NONE</u> , *PRINT	Optional
RLS	Release	<i>Character value</i> , * <u>FIRST</u>	Optional
REPLACERLS	Replace release	<i>Character value</i> , * <u>ONLY</u> , *NO	Optional
VOL	Volume identifier	Single values: * <u>MOUNTED</u> Other values (up to 75 repetitions): <i>Character value</i>	Optional

Keyword	Description	Choices	Notes
SEQNBR	Sequence number	1-16777215, <u>*SEARCH</u>	Optional
ENDOPT	End of media option	<u>*REWIND</u> , *LEAVE, *UNLOAD	Optional
SAVF	Save file	<i>Qualified object name</i>	Optional
	Qualifier 1: Save file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
LIB	Library	<i>Element list</i>	Optional
	Element 1: Library 1	<i>Name</i> , <u>*SAME</u>	
	Element 2: Library 2	<i>Name</i> , <u>*SAME</u>	
	Element 3: Library 3	<i>Name</i> , <u>*SAME</u>	
	Element 4: Library 4	<i>Name</i> , <u>*SAME</u>	
	Element 5: Library 5	<i>Name</i> , <u>*SAME</u>	
	Element 6: Library 6	<i>Name</i> , <u>*SAME</u>	
	Element 7: Library 7	<i>Name</i> , <u>*SAME</u>	
	Element 8: Library 8	<i>Name</i> , <u>*SAME</u>	
	Element 9: Library 9	<i>Name</i> , <u>*SAME</u>	
	Element 10: Library 10	<i>Name</i> , <u>*SAME</u>	
	Element 11: Library 11	<i>Name</i> , <u>*SAME</u>	
LNLIB	Language library	<i>Name</i> , <u>*SAME</u>	Optional
FLR	Folder	<i>Name</i> , <u>*SAME</u>	Optional
CODHOMEDIR	Code home directory	Single values: *PROMPT Other values (up to 300 repetitions): <i>Path name</i> , <u>*SAME</u>	Optional
LNGHOMEDIR	Language home directory	Single values: *PROMPT Other values (up to 300 repetitions): <i>Path name</i> , <u>*SAME</u>	Optional
FRCOBJCVN	Force object conversion	Single values: <u>*SYSVAL</u> , *NO Other values: <i>Element list</i>	Optional
	Element 1: Convert during restore	<u>*YES</u>	
	Element 2: Objects to convert	<u>*RQD</u> , *ALL	

Top

---

## Product (LICPGM)

Specifies the seven-character identifier of the licensed program that is restored.

This is a required parameter.

Top

---

## Device (DEV)

Specifies the name of the device used for the restore of the licensed program. The device name must already be known on the system by a device description. Use the Work with Device Descriptions (WRKDEV) command to display the names of the devices available on this system.

This is a required parameter.

The possible values are:

**\*SAVF** The restore operation is done using the save file name specified on the save file (SAVF) parameter.

***optical-device-name***

Specify the name of the optical device used for the restore operation.

***tape-media-library-device-name***

Specify the name of the tape media library device used for the restore operation.

***tape-device-name***

Specify the name of one or more tape devices to use to restore the licensed program. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Top

---

## Optional part to be restored (OPTION)

Specifies which one of the optional parts of the licensed program given in the **Product** prompt (LICPGM parameter) is to be restored.

The possible values are:

**\*BASE**

Only the base part of the licensed program is to be restored.

***number-of-licensed-program-option***

Specify the number of the optional part of the listed licensed program that is to be restored.

Top

---

## Type of object to be restored (RSTOBJ)

Specifies the type of licensed program objects to be restored.

The possible values are:

**\*ALL** All of the objects for the licensed program are restored. This includes both program objects and the language objects specified on the **Language for licensed program** prompt (LNG parameter).

If a tape device is specified on the DEV parameter, then the RSTOBJ(\*ALL) value is used when the saving of the licensed program has been done with the SAVLICPGM command such that the language objects immediately follow the program objects on the tape media. If the language objects (\*LNG) and programming objects (\*PGM) are not in consecutive order on the distribution tape, \*ALL cannot be used in most cases. Instead, the program and language objects must be restored separately. The DSPTAP command can be used to determine the order of the objects on the tape. An example of how to restore language and program objects separately is in the "Examples" section at the end of this command.

If \*SAVF is specified on the DEV parameter, then the RSTOBJ(\*ALL) value can be used when the saving of the licensed program has been done with the SAVLICPGM command using OBJTYPE(\*ALL).

**\*PGM** Only the program objects for the licensed program are restored. \*PGM should be used when restoring program objects from a distribution media where the program objects and selected language objects are not on the same distribution media or are not in consecutive order.

\*LNG Only the language objects for a licensed program are restored. The **Language for licensed program** prompt (LNG parameter) determines if the objects are associated with the licensed program library or with a multilingual library.

Top

---

## Language for licensed program (LNG)

Specifies the national language version (NLV) to be used for restoring the licensed program. If the language feature of the licensed program on the save media matches the system language feature, the language objects are restored to the licensed program's libraries. If the language features do not match, the language objects are restored into the multilingual library for that language feature.

The possible values are:

### \*PRIMARY

The language feature of the operating system is restored for the specified licensed program.

**Note:** Use GO LICPGM with option 20 to display the primary language of the operating system.

### \*SAVVOL

The language file on the mounted volume is to be restored for the licensed program. This option is not valid with DEV(\*SAVF).

### *feature-codes*

Specify the NLV identifier for the language file that is to be restored for the licensed program. More information on feature identifications and a list of IBM-supplied feature codes is in the Install, upgrade, or delete i5/OS and related software book, SC41-5120.

Top

---

## Output (OUTPUT)

Specifies whether a listing that shows information about the status of the objects is created. The listing shows the parameter values that were specified and shows all objects, restored and not restored. Information about each object's security is listed for the restored objects.

The possible values are:

### \*NONE

No output is created.

### \*PRINT

The output is printed with the job's spooled output.

Top

---

## Release (RLS)

Specifies the version, release, and modification level of the licensed program being restored.

The possible values are:

### \*FIRST

The first version, release, and modification level found on the distribution media is restored.

### *release-level*

Specify the release level in VxRyMz format, where Vx is the version number, Ry is the release



number, and Mz is the modification level. The variables x and y can be a number from 0 through 9, and the variable z can be a number from 0 through 9 or a letter from A through Z.

Top

---

## Replace release (REPLACERLS)

Specifies the version, release, and modification level of the licensed program being replaced.

The possible values are:

### \*ONLY

Replace only the version, release, and modification level of the licensed program currently installed.

**\*NO** The licensed program currently installed on the system is not replaced. The licensed program being restored must be a different release than the one currently installed. If the licensed program being restored exists in the same libraries as the installed program, an override parameter must be specified indicating to which libraries the licensed program is restored.

### *release-level*

Specify the release level in VxRyMz format, where Vx is the version number, Ry is the release number, and Mz is the modification level. The variables x and y can be a number from 0 through 9, and the variable z can be a number from 0 through 9 or a letter from A through Z.

Top

---

## Volume identifier (VOL)

Specifies the volume identifiers of the media or the cartridge identifiers of tapes in a tape media library device, from which the objects are being restored. The volumes must be in the same order as they were when the data was saved. The volume that contains the beginning of the file to be restored should be placed in the device.

The possible values are:

### \*MOUNTED

The licensed program is restored from the volumes that are currently on the device specified on the **Device** prompt (DEV parameter). For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

**Note:** This value cannot be specified when using an optical media library device.

### *volume-identifier*

Specify the identifiers of one or more volumes in the order they are put on the device and used to restore the licensed program. A maximum of 75 volume identifiers can be specified.

Top

---

## Sequence number (SEQNBR)

Specifies which sequence number to use for the restore process. This option is only valid for tape devices.

The possible values are:

### \*SEARCH

The volume that is placed in the device is searched for a data file with an identifier that matches the label for the specified licensed program or licensed program option. When a match is found,

the objects are restored. If the last operation on the device specified **\*LEAVE** on the **End of media option** prompt (ENDOPT parameter), indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If **\*LEAVE** was not used for the **End of media option** prompt (ENDOPT parameter) of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

*file-sequence-number*

Specify the sequence number of the file that is used for the restore process. Valid values range from 1 through 16777215.

Top

---

## End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the restore operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

**Note:** This parameter is valid only if a tape or optical device name is specified for the DEV parameter. For optical devices, **\*UNLOAD** is the only special value supported, **\*REWIND** and **\*LEAVE** will be ignored.

**\*REWIND**

The tape is automatically rewound, but not unloaded, after the operation has ended.

**\*LEAVE**

The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

**\*UNLOAD**

The tape is automatically rewound and unloaded after the operation ends. Some optical devices will eject the volume after the operation ends.

Top

---

## Save file (SAVF)

Specifies the qualified name of the save file containing the product.

The name of the save file can be qualified by one of the following library values:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*library-name*

Specify the name of the library where the save file exists.

The possible values are:

*save-file-name*

Specify the name of the save file.

Top

---

## Library (LIB)

Specifies the libraries into which the licensed program is being restored. This function is not supported by all licensed programs.

The possible values are:

\*SAME

The licensed program is restored into the specified library.

*library-name*

Specify the name of the library into which the licensed program is being restored. A maximum of 11 libraries can be specified.

Top

---

## Language library (LNLIB)

Specifies the secondary language library into which the secondary language objects are restored. This function is not supported by all licensed programs.

The possible values are:

\*SAME

The licensed program is restored into the specified secondary language library.

*library-name*

Specify the name of the secondary language library into which the licensed program is restored.

Top

---

## Folder (FLR)

Specifies the name of the root folder into which the licensed program is being restored. This function is not supported by all licensed program.

The possible values are:

\*SAME

Use the specified root folder.

*folder-name*

Specify the name of the root folder. The root folder is the folder on the system containing all of the other folders.

Top

---

## Code home directory (CODHOMEDIR)

Specifies the directories into which the code part of the product is being restored. This function is not supported by all products.

**Note:** This parameter is mutually exclusive with the FLR parameter.

The possible values are:

\*SAME

The code part of the product is restored into the directories specified when packaged or already

installed. \*SAME may be specified as the only parameter value or within a list of directories. If used within a list, \*SAME specifies that a particular directory is unchanged, though other directories may be different than when the product was packaged or previously installed.

**\*PROMPT**

The code directories to be used are displayed. If the product is not currently installed, the directory names can be changed.

*path-name*

Specify the home path directory name into which the code part of the product is being restored. Up to 300 directories may be specified. For directory name entries which are unchanged, \*SAME can be specified for the path name.

Top

---

## Language home directory (LNGHOMEDIR)

Specifies the directories into which the language part of the product is being restored. This function is not supported by all products.

**Note:** This parameter is mutually exclusive with the FLR parameter.

The possible values are:

**\*SAME**

The language part of the product is restored into the directories specified when packaged or already installed. \*SAME may be specified as the only parameter value or within a list of directories. If used within a list, \*SAME specifies that a particular directory is unchanged, though other directories may be different than when the product was packaged or previously installed.

**\*PROMPT**

The language directories to be used are displayed. If the product is not currently installed, the directory names can be changed.

*path-name*

Specify the home path directory name into which the language part of the product is being restored. Up to 300 directories may be specified. For directory name entries which are unchanged, \*SAME can be specified for the path name.

Top

---

## Force object conversion (FRCOBJCVN)

Specifies whether to convert user objects to the format required for use in the current version of the operating system when the objects are restored.

**Notes:**

1. This parameter applies only to user objects of the \*MODULE, \*PGM, \*SRVPGM, and \*SQLPKG object types.
2. An object must have creation data (either observable or unobservable) to be converted.
3. If an object needs to be converted (because it is formatted for an earlier version of the operating system), but is not converted during this restore operation, the object is automatically converted the first time it is used.

**Single values**

### \*SYSVAL

The objects are converted based on the value of the QFRCCVNRST system value.

**\*NO** The objects are not converted during the restore operation.

**Note:** If FRCOBJCVN(\*NO) is specified, then the QFRCCVNRST system value must have a value of either "0" or "1".

### **Element 1: Convert during restore**

**\*YES** The objects are converted during the restore operation.

#### **Notes:**

1. If FRCOBJCVN(\*YES \*RQD) is specified, then the QFRCCVNRST system value must have a value of "0", "1", or "2". FRCOBJCVN(\*YES \*RQD) will override a QFRCCVNRST value of "0" or "1". If FRCOBJCVN(\*YES \*ALL) is specified, then QFRCCVNRST can have any valid value and FRCOBJCVN(\*YES \*ALL) overrides the QFRCCVNRST system value.
2. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

### **Element 2: Objects to convert**

**\*RQD** The objects are converted only if they require conversion to be used by the current operating system. If the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

**\*ALL** All objects are converted regardless of their current format, including objects already in the current format. However, if the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

Top

---

## **Examples**

### **Example 1: Restoring Program Using Defaults**

```
RSTLICPGM LICPGM(5722WDS) DEV(TAP01)
```

This command restores the WebSphere Development Studio (5722WDS) licensed program to the system. The tape containing the licensed program objects must be put on the TAP01 tape drive. Because no other parameters are specified, the defaults are used for the command.

### **Example 2: Restoring a Third Version of a Product**

```
RSTLICPGM LICPGM(1MYPROD) OPTION(*BASE)
DEV(TAP01) RLS(V5R2M0)
REPLACERLS(*NO) LIB(A B C)
```

This command restores the base part of the V5R2M0 1MYPROD product to the system if the base of the V5R2M0 1MYPROD product is not currently installed on the system.

### **Example 3: Restoring One Version of a Product Over Another Version**

```
RSTLICPGM LICPGM(2MYPROD) OPTION(*BASE)
DEV(TAP01) RLS(*FIRST)
REPLACERLS(*ONLY)
```

This command restores the first version release modification level of the base part of the 2MYPROD product that is found on the tape in the TAP01 drive. It does not matter if the version release modification level of the base of the product on the tape matches the version release modification level of the base of the product on the system.

#### Example 4: Restoring Product From Save File

```
RSTLICPGM LICPGM(5722WDS) DEV(*SAVF) SAVF(MYLIB/MYSAVF)
```

This command restores the WebSphere Development Studio (5722WDS) licensed program to the system from the save file MYSAVF in MYLIB. Because no other parameters are specified, the defaults are used for the command.

#### Example 5: Restoring a Third Version of a Product From a Save File

```
RSTLICPGM LICPGM(1MYPROD) OPTION(*BASE)  
          DEV(*SAVF) RLS(V5R2M0)  
          REPLACERLS(*NO) LIB(A B C) SAVF(MYLIB/MYSAVF)
```

This command restores the base part of the V5R2M0 1MYPROD product to the system from save file MYSAVF in MYLIB if the base of the V5R2M0 1MYPROD product is not currently installed on the system.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF3D94

No product found in save file.

#### CPF3D96

Objects for product &1 option &2 release &4 not restored.

#### CPF37A2

Installation cannot be performed.

#### CPF3728

Device &1 specified with other devices.

#### CPF3733

&2 &1 in &3 previously damaged.

#### CPF3739

Database file &1 member in &3 damaged.

#### CPF3820

&4 objects for &1 option &2 not restored.

#### CPF3880

No language objects exist.

#### CPF3884

Licensed program &1 option &2 not processed.

#### CPI36C9

Error occurred while removing PTFs.

Top

---

## Restore Object (RSTOBJ)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Restore Object (RSTOBJ) command restores to the system a single object or a group of objects in a single library that were saved on a tape or optical volume, or in a save file using a single command. Objects to be restored could have been saved by the Save Object (SAVOBJ), Save Changed Objects (SAVCHGOBJ) or Save Library (SAVLIB) command. The RSTOBJ command restores the object description and contents of each object specified in the command.

For job queues, user-defined message queues, logical files, and user queues, only the object descriptions are restored; the contents of those objects are not restored. However, logical file access paths can be saved by a save command with ACCPTH(\*YES) specified; if this is done, the access paths can be restored. More information on restoring access paths is in the Database information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>. The contents of spooled files on output queues can be restored, if they were saved, by specifying \*NEW for the **Spooled file data (SPLFDTA)** parameter.

The command can be used to restore the objects even if the object storage was freed when the objects were saved, or if the objects were deleted after they were saved. If the storage was not freed as part of a save operation, each object is restored in the same area of storage that it previously occupied. If the version of the object being restored is larger than the version in the system (for example, data records that were deleted from the system still exist in the saved version of a file), the additional storage needed for the object is acquired. If the saved version of the object is smaller (for example, data records that are added to the system), the space that was acquired for the object remains assigned to that object and available for use by that object.

If logical file access paths were saved (ACCPTH(\*YES) was specified when the objects were saved), the access paths are restored if (1) all based-on physical files are also being restored by the same restore command, (2) the logical file is also being restored by the same restore command, or the logical file already exists on the system (the same file exists, not a re-created version), and (3) MAINT(\*IMMED or \*DLY) is in effect for the logical file if it still exists on the system.

If the storage was freed, the system finds the storage space needed to store the contents (only the data portion) of each file, module, program, service program, journal receiver, and Structured Query Language (SQL) package. If the objects do not exist on the system because they were deleted or they are being restored in a different system, the system must find the storage to store everything (the description and the data portion) about each unknown object.

The user profile of the system default owner (QDFTOWN) becomes the default owner of objects restored in the system whose owner is not known to the system.

If an object is being restored over an existing object on the system, the object auditing value of the existing object is kept. If the object is being restored as new to the system, the object auditing value is restored from the media.

**ATTENTION:** Do not use this command to restore licensed programs into library QSYS. Unpredictable results can occur.

### Restrictions:

- This command is shipped with no public authority (\*EXCLUDE).

- You must have use (\*USE) authority for the Create Save File (CRTSAVF) command when restoring a save file that does not currently exist on the system.
- When saving or restoring to an existing database file using the OUTFILE parameter, you must have execute (\*EXECUTE) authority to the library where the output database file is located.
- With the exception of overrides for the restore listing file and database files specified for the OUTFILE parameter, this command ignores all file overrides that are currently in effect for the job.
- You must have save system (SAVSYS) special authority, or all of the following object authorities:
  - Add (\*ADD) and execute (\*EXECUTE) authorities for the specified library and save file library.
  - Object existence (\*OBJEXIST) authority for, or be the owner of, each object specified if the object already exists in the library on the system. \*OBJEXIST and use (\*USE) authorities are required for message queue objects. If the object does not already exist in the library on the system, you must have \*ADD authority for the user profiles that own each object being restored. For spooled file data, \*OBJEXIST authority is required for the output queue to which it is restored.
  - If VOL(\*SAVVOL) is specified, \*USE authority to the saved-from library.
- The RSTOBJ command does not restore the library's data dictionary or its associated database files. To do this, the RSTLIB command must be used.
- You must have \*USE authority for the save file to restore from the save file. In addition, the user must have \*USE authority for the device description when restoring from a tape or optical device.
- When using a media definition, you must have \*USE authority to the media definition and \*EXECUTE authority to the media definition library.
- If this command is used to restore an OPM program, the copy of that program that is currently in the system must not be running while the program is being restored. If this occurs, the running program may fail or behave unpredictably.
- Objects saved by separate commands must also be restored by separate commands. If a single command is used, some of the objects are not restored.
- If the user domain object user space (\*USRSPC), user index (\*USRIDX), or user queue (\*USRQ) is restored to a library that is not permitted in the system value QALWUSRDMN (allow user domain objects in libraries), the object is converted to a system domain object.

Top

---

## Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
SAVLIB	Saved library	Single values: *ANY Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 2
DEV	Device	Single values: *SAVF, *MEDDFN Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 3



Keyword	Description	Choices	Notes
OBJTYPE	Object types	Single values: <b>*ALL</b> Other values (up to 74 repetitions): *ALRTBL, *BNDDIR, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *NWSCFG, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMORY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *TIMZON, *USRIDX, *USRQ, *USRSPC, *VLDL, *WSCST	Optional, Positional 4
VOL	Volume identifier	Single values: <b>*MOUNTED</b> , *SAVVOL Other values (up to 75 repetitions): <i>Character value</i>	Optional, Positional 5
SEQNBR	Sequence number	1-16777215, <b>*SEARCH</b>	Optional, Positional 6
LABEL	Label	<i>Character value</i> , <b>*SAVLIB</b>	Optional
ENDOPT	End of media option	<b>*REWIND</b> , *LEAVE, *UNLOAD	Optional
SAVF	Save file	<i>Qualified object name</i>	Optional
	Qualifier 1: Save file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b> , *CURLIB	
OPTION	Option	<b>*ALL</b> , *NEW, *OLD, *FREE	Optional
FILEMBR	File member	Values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: File	<i>Name</i> , <b>*ALL</b>	
	Element 2: Member	Single values: <b>*ALL</b> , *NONE Other values (up to 50 repetitions): <i>Generic name, name</i>	
MBROPT	Data base member option	<b>*MATCH</b> , *ALL, *NEW, *OLD	Optional
SPLFDTA	Spooled file data	<b>*NEW</b> , *NONE	Optional
SAVDATE	Date when saved	<i>Date</i>	Optional
SAVTIME	Time when saved	<i>Time</i>	Optional
ALWOBJDIF	Allow object differences	Single values: <b>*NONE</b> , *ALL Other values (up to 4 repetitions): *AUTL, *FILELVL, *OWNER, *PGP	Optional
FRCOBJCVN	Force object conversion	Single values: <b>*SYSVAL</b> , *NO Other values: <i>Element list</i>	Optional
	Element 1: Convert during restore	<b>*YES</b>	
	Element 2: Objects to convert	<b>*RQD</b> , *ALL	
RSTLIB	Restore to library	<i>Name</i> , <b>*SAVLIB</b>	Optional
OUTPUT	Output	<b>*NONE</b> , *PRINT, *OUTFILE	Optional
MEDDFN	Media definition	<i>Qualified object name</i>	Optional
	Qualifier 1: Media definition	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b> , *CURLIB	
OPTFILE	Optical file	<i>Path name</i> , <b>*</b>	Optional
RSTASPDEV	Restore to ASP device	<i>Name</i> , <b>*SAVASPDEV</b>	Optional
RSTASP	Restore to ASP number	1-32, <b>*SAVASP</b>	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	

Keyword	Description	Choices	Notes
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	<i>*REPLACE, *ADD</i>	
INFTYPE	Type of output information	<i>*OBJ, *MBR</i>	Optional
OMITLIB	Libraries to omit	Single values: <i>*NONE</i> Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional
OMITOBJ	Objects to omit	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Generic name, name, *NONE, *ALL</i>	
	Qualifier 2: Library	<i>Generic name, name, *ALL</i>	
	Element 2: Object type	<i>Character value, *ALL</i>	

Top

---

## Objects (OBJ)

Specifies the names of one or more objects to restore. The objects you specify will be restored from the first file that meets the search value and contains any of the objects. If a tape contains multiple files for the same library, you may need to specify the tape file sequence number to ensure that all of the correct objects are restored.

If the file does not contain all of the objects you specify, diagnostic messages will be issued for the objects not found. The completion message will contain a count of objects restored and objects not restored.

If the **Object types (OBJTYPE)** parameter is not specified when the command is run, all the object types listed in the description of that parameter are restored, if they are in the specified library on the media or in the save file, and if they have the specified names.

This is a required parameter.

### Single values

**\*ALL** All the objects saved from the specified library are restored, depending on the values specified for the **Object types (OBJTYPE)** and **Option (OPTION)** parameters.

### Other values (up to 300 repetitions)

#### *generic-name*

Specify one or more generic names of groups of objects in the specified library to restore. A generic name is a character string that contains one or more characters followed by an asterisk (\*). If an asterisk is not specified with the name, the system assumes that the name is a complete object name.

**name** Specify one or more names of specific objects to restore. Both generic names and specific names can be specified in the same command.

Top

---

## Saved library (SAVLIB)

Specifies the libraries that contained the saved objects. If \*SAVLIB is specified or defaulted for the **Restore to library (RSTLIB)** parameter, this is also the name of the library to which the objects are restored.

### Notes:

1. If you specify more than one library, or a single value, or a generic library value:
  - The **Device (DEV)** parameter must not specify \*SAVF.
  - The **Restore to library (RSTLIB)** parameter must specify \*SAVLIB.
  - The **Label (LABEL)** parameter must specify \*SAVLIB.
  - The **Optical file (OPTFILE)** parameter must specify \* or end with /\*.
2. The tape or optical file that you select to restore may contain a library that does not match the value that you specify for this parameter. This parameter is used to determine the default LABEL value for a tape file and the default OPTFILE value for an optical file, but the restore operation does not verify that this parameter matches the saved library.

This is a required parameter.

### Single values

**\*ANY** Restores objects from the first version of all saved libraries found on the tape beginning with the sequence number specified for the **Sequence number (SEQNBR)** parameter, or restores objects from all saved libraries found on the optical media in the directory specified for the OPTFILE parameter.

### Other values (up to 300 repetitions)

#### *generic-name*

Specify the generic name of the libraries to be restored.

**Note:** A generic name is specified as a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

**name** Specify the name of the library to restore. The name of the library being restored must be the same as the name that was used when the library was saved.

Top

---

## Device (DEV)

Specifies the name of the device used for the restore operation. The device name must already be known on the system by a device description.

This is a required parameter.

### Single values

**\*SAVF** The restore operation is done using the save file specified for the **Save file (SAVF)** parameter.

#### **\*MEDDFN**

The restore operation is done using the devices and media identified in the media definition specified for the **Media definition (MEDDFN)** parameter.

### Other values

*optical-device-name*

Specify the name of the optical device used for the restore operation.

*tape-media-library-device-name*

Specify the name of the tape media library device used for the restore operation.

*tape-device-name*

Specify the names of one or more tape devices used for the restore operation. If a virtual tape device is used, it must be the only device specified. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order in which they are used. When more than one tape volume is to be restored, using more than one tape device permits one tape volume to be rewound while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Top

---

## Object types (OBJTYPE)

Specifies the types of system objects that are restored.

### Single values

**\*ALL** All object types that are specified by name and were saved from the specified library are restored. If \*ALL is also specified for the **Objects (OBJ)** parameter, all objects saved for that library are restored.

### Other values

*object-type*

Specify the value for each of the types of objects to be restored, such as command (\*CMD), file (\*FILE), or program (\*PGM).

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

**Note:** \*CFGL object types can be restored using the Restore Object (RSTOBJ) command only from objects saved on releases prior to V2R2M0. \*CFGL object types saved on V2R2M0 and newer releases are restored using the Restore Configuration (RSTCFG) command.

Top

---

## Volume identifier (VOL)

Specifies the volume identifiers of the media or the cartridge identifiers of tapes in a tape media library device, from which the objects are being restored. The volumes must be in the same order as they were when the data was saved. The volume that contains the beginning of the file to be restored should be placed in the device.

**Note:** The version of the objects restored is the first version found in the specified location, unless a specific version is identified by the **Date when saved (SAVDATE)** and **Time when saved (SAVTIME)** parameters.

### Single values

**\*MOUNTED**

The objects are restored from the volumes placed in the device specified for the **Device (DEV)**

parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

**Note:** This value cannot be specified when using an optical media library device.

#### **\*SAVVOL**

The system, by using the save or restore history information, determines which volumes contain the most recently saved version of the objects.

**Note:** If this is a restore from an optical device, only the first 6 characters of the volume name are available. If the volume name of the optical media exceeds 6 characters, the volume may not be found. You should specify the complete volume or file name on the command instead of using VOL(\*SAVVOL) when the name of the optical media exceeds 6 characters.

When \*SAVVOL is specified, the following operational characteristics and restrictions apply:

- If the characteristics of the device specified in the DEV parameter do not match the device and location of the most recently saved version of the library, an error message is sent to the **user**, and the function is ended.
- If the wrong volume is placed in a device in the location specified by the command, a message is sent to the **system operator** that identifies the first volume that must be placed in the device before the objects can be restored.
- When OBJ(\*ALL) and OBJTYPE(\*ALL) is specified, the objects are considered in the order they would appear in a display produced by the Display Library (DSPLIB) command. The object names and types specified in the RSTOBJ command are used to determine which file of saved objects is used in the restore operation. One file is produced for each SAVLIB or SAVOBJ command run. The file chosen is the one in which the first considered object was last saved. Objects that were not saved in the file chosen to be processed, or that were more recently saved, are not restored; an error message is sent to the user for each unrestored object.
- If \*SAVVOL is specified, the SAVDATE and SAVTIME parameters cannot be specified.
- If \*SAVVOL is specified and the RSTLIB value is equal to the SAVLIB value, OPTION(\*NEW) cannot be specified.
- If \*SAVVOL is specified, SEQNBR(\*SEARCH) and LABEL(\*SAVLIB) must be specified.

#### **Other values (up to 75 repetitions)**

##### *character-value*

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to restore the data.

Top

---

## **Sequence number (SEQNBR)**

Specifies, when tape is used, which sequence number is used for the restore operation.

#### **\*SEARCH**

The volume in the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the object is restored. If the last operation on the device specified \*LEAVE for the **End of tape option (ENDOPT)** parameter, indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If \*LEAVE was not used for the **End of tape option (ENDOPT)** parameter of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

#### **1-16777215**

Specify the sequence number of the file to be used for the restore operation.

---

## Label (LABEL)

Specifies the name that identifies the data file to be used for the restore operation. This label must have been specified on the save command.

### \*SAVLIB

The file label is the name specified for the **Saved library (SAVLIB)** parameter.

### *character-value*

Specify the data file identifier of the data file used for the restore operation. A maximum of 17 characters can be used. This option is valid only for a single-library restore.

Top

---

## End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the restore operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

**Note:** This parameter is valid only if a tape or optical device name is specified for the DEV parameter. For optical devices, \*UNLOAD is the only special value supported, \*REWIND and \*LEAVE will be ignored.

### \*REWIND

The tape is automatically rewound, but not unloaded, after the operation has ended.

### \*LEAVE

The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

### \*UNLOAD

The tape is automatically rewound and unloaded after the operation ends. Some optical devices will eject the volume after the operation ends.

Top

---

## Save file (SAVF)

Specifies the save file used to restore the data.

**Note:** A value must be specified for this parameter if \*SAVF is specified for the **Device (DEV)** parameter.

### Qualifier 1: Save file

*name* Specify the name of save file to be used.

### Qualifier 2: Library

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

### \*CURLIB

The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

*name* Specify the name of the library where the save file is located.

---

## Option (OPTION)

Specifies how to handle restoring each object.

- \*ALL** All the objects in the saved library are restored to the library. Objects in the saved library replace the current versions in the system library. Objects not having a current version are added to the system library. Objects presently in the library, but not on the media, remain in the library.
- \*NEW** Only the objects in the saved library that do not exist in the current version of the system library are added to the library. Only objects not known to the system library are restored; known objects are not restored. This option restores objects that were deleted after they were saved or that are new to this library. If any saved objects have a version already in the system library, they are not restored, and an informational message is sent for each one, but the restore operation continues.
- \*OLD** Only the objects in the library having a saved version are restored; that is, the version of each object currently in the library is replaced by the saved version. Only objects known to the library are restored. If any saved objects are no longer part of the online version of the library, they are not added to the library; an informational message is sent for each one, but the restore continues.
- \*FREE** The saved objects are restored only if they exist in the system library with their space freed. The saved version of each object is restored on the system in its previously freed space. This option restores objects that had their space freed when they were saved. If any saved objects are no longer part of the current version of the library, or if the space is not free for any object, the object is not restored and an informational message is sent for each one. The restore operation continues, and all of the freed objects are restored.

Top

---

## File member (FILEMBR)

Specifies the database file members to restore. This parameter is made up of two parts: the file name and the member name. This parameter cannot be specified if \*MATCH is specified for the **Data base member option (MBROPT)** parameter.

Each database file specified here must also be specified for the **Objects (OBJ)** parameter, by a specific name, a generic name, or \*ALL. The **Object types (OBJTYPE)** parameter must either be \*ALL or include \*FILE. Generic names are not valid for the database file name but are allowed for the member name. Duplicate file names are not allowed.

There may be up to 50 of the file/member element list combinations specified for this parameter.

### Element 1: File

**\*ALL** The list of member name values that follows this value applies to all files indicated on the OBJ parameter.

*name* Specify the name of the database file from which the listed members are restored.

### Element 2: Member

#### Single values

**\*ALL** All members are restored from the specified file.

**\*NONE**

No members are restored from the specified file. Only the file description is restored.



## Other values (up to 50 repetitions)

### *generic-name*

Specify the generic name of the members to be restored from the specified file. A generic name is a character string that contains one or more characters followed by an asterisk (\*). If an asterisk is not specified with the name, the system assumes that the name is a complete member name.

*name* Specify the name of the member to be restored from the specified file.

Top

---

## Data base member option (MBROPT)

Specifies, for database files that exist on the system, which members are restored. If \*MATCH is used, the member list in the saved file must match, member for member, the current version on the system. All members are restored for files that do not exist, if the file is restored.

### \*MATCH

The saved members are restored if the lists of the members where they exist match, member for member, the lists of the current system version. MBROPT(\*MATCH) is not valid when \*ALL is specified for the **Allow object differences (ALWOBJDIF)** parameter.

\*ALL All members in the saved file are restored.

\*NEW Only new members (members not known to the system) are restored.

\*OLD Only members already known to the system are restored.

Top

---

## Spooled file data (SPLFDTA)

Specifies whether to restore spooled file data and attributes.

\*NEW For each output queue that is restored, spooled file data that was saved with the output queue is restored, if it does not already exist on the system.

### \*NONE

No spooled file data is restored.

Top

---

## Date when saved (SAVDATE)

Specifies the date when the objects were saved. If the most recently saved version is not the one being restored, or if multiple saved versions exist on the media volume, specify the date that indicates which version of the objects to restore. If a volume identifier or \*MOUNTED is specified for the **Volume identifier (VOL)** parameter, but SAVDATE is not, the version of the objects that is restored is the first version found on the volume. This parameter is valid only if a volume identifier or \*MOUNTED is specified for the VOL parameter or if a save file is specified for the **Save file (SAVF)** parameter. This parameter is ignored when the **Sequence number (SEQNBR)** parameter is specified.

*date* Specify the date that the library to be restored was saved. The date must be entered in the job date format.

Top



---

## Time when saved (SAVTIME)

Specifies the time at which the objects were saved.

If a volume identifier or \*MOUNTED is specified for the **Volume identifier (VOL)** parameter, but this parameter is not specified, the version of the objects to be restored is the first version found on the volume. This parameter is valid only if the **Date when saved (SAVDATE)** parameter is also specified.

This parameter is ignored when a number is specified for the **Sequence number (SEQNBR)** parameter.

- time* Specify the time that the library to be restored was saved. The time can be specified with or without a time separator:
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
  - With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Top

---

## Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

### Notes:

1. You must have all object (\*ALLOBJ) special authority to specify any value other than \*NONE for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- **Authorization list:** The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.  
**Note:** This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.
- **File level id:** The creation date and time of the database file on the system does not match the creation date and time of the file that was saved.
- **Member level id:** The creation date and time of the database file member on the system does not match the creation date and time of the member that was saved.
- **Ownership:** The owner of an object on the system is different than the owner of an object from the save operation.
- **Primary Group:** The primary group of an object on the system is different than the primary group of an object from the save operation.

### Single values

#### \*NONE

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

**\*ALL** All of the differences listed above are allowed on the restore operation. File level id and member

level id differences are handled differently than the \*FILELVL value. If there is a file level difference and \*ALL is specified for the **Data base member option (MBROPT)** parameter, the existing version of the file is renamed and the saved version of the file is restored. If there is a member level difference, the existing version of the member is renamed and the saved version of the member is restored. This value will restore the saved data, but the result may not be correct. You will need to choose whether the restored data or the renamed data is correct, and you will need to make the necessary corrections to the database. For other differences, see the description of each individual value to determine how differences are handled.

#### **Other values (up to 4 repetitions)**

##### **\*AUTL**

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to \*EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to \*EXCLUDE.

##### **\*FILELVL**

File level id and member level id differences are allowed. An attempt will be made to restore existing physical files even though the physical file on the save media may have a different file level id or member level id than the physical file on the system. The physical file data will only be restored for those physical files whose format level identifiers on the save media match the format level identifiers of the corresponding physical file on the system.

If this value is not specified, file level id and member level id differences are not allowed. If an object already exists on the system with a different file level id or member level id than the saved object, the object is not restored.

##### **\*OWNER**

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

##### **\*PGP**

Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Top

---

## **Force object conversion (FRCOBJCVN)**

Specifies whether to convert user objects to the format required for use in the current version of the operating system when the objects are restored.

#### **Notes:**

1. This parameter applies only to user objects of the \*MODULE, \*PGM, \*SRVPGM, and \*SQLPKG object types.
2. An object must have creation data (either observable or unobservable) to be converted.
3. If an object needs to be converted (because it is formatted for an earlier version of the operating system), but is not converted during this restore operation, the object is automatically converted the first time it is used.

### Single values

#### \*SYSVAL

The objects are converted based on the value of the QFRCCVNRST system value.

**\*NO** The objects are not converted during the restore operation.

**Note:** If FRCOBJCVN(\*NO) is specified, then the QFRCCVNRST system value must have a value of either "0" or "1".

### Element 1: Convert during restore

**\*YES** The objects are converted during the restore operation.

#### Notes:

1. If FRCOBJCVN(\*YES \*RQD) is specified, then the QFRCCVNRST system value must have a value of "0", "1", or "2". FRCOBJCVN(\*YES \*RQD) will override a QFRCCVNRST value of "0" or "1". If FRCOBJCVN(\*YES \*ALL) is specified, then QFRCCVNRST can have any valid value and FRCOBJCVN(\*YES \*ALL) overrides the QFRCCVNRST system value.
2. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

### Element 2: Objects to convert

**\*RQD** The objects are converted only if they require conversion to be used by the current operating system. If the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

**\*ALL** All objects are converted regardless of their current format, including objects already in the current format. However, if the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

Top

---

## Restore to library (RSTLIB)

Specifies whether the objects are restored to a different library or to the same library where they were saved.

#### \*SAVLIB

The objects are restored to the same library from which they were saved.

**name** Specify the name of the library to which the saved objects are restored.

Top

---

## Output (OUTPUT)

Specifies whether a listing that shows information about the status of the objects is created and directed to an output file. The listing shows the restore information and shows all objects restored, not restored, and excluded. Information about each object's security is listed for the restored objects.

**\*NONE**

No output is created.

**\*PRINT**

The output is printed with the job's spooled output.

**\*OUTFILE**

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

**Note:** You must specify a database file name for the OUTFILE parameter when OUTPUT(\*OUTFILE) is specified.

Top

---

## Media definition (MEDDFN)

Specifies the media definition (\*MEDDFN) object that identifies the devices and media used to restore the data. For information about creating and using a media definition, see the Backup and Recovery book, SC41-5304, and the Create Media Definition API in the System API Reference information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

If a media definition is specified, the VOL, SEQNBR, SAVE, and OPTFILE parameters cannot be specified. The volume identifiers and sequence numbers are specified in the media definition.

### Qualifier 1: Media definition

*name* Specify the name of the media definition to be used.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched.

*name* Specify the name of the library to be searched.

Top

---

## Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume.

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

**\*** The system searches the root directory of the optical volume for the default name generated by the corresponding save operation.

**'optical-directory-path-name\*'**

The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

**'optical-file-path-name'**

Specify the path name of the optical file.

---

## Restore to ASP device (RSTASPDEV)

Specifies the auxiliary storage pool (ASP) device to which the data is to be restored.

**Note:** You can specify either the RSTASPDEV parameter or the RSTASP parameter, but not both.

### \*SAVASPDEV

The data is restored to the same ASP from which it was saved.

*name* Specify the name of the ASP device to be used.

---

## Restore to ASP number (RSTASP)

Specifies whether objects are restored to the auxiliary storage pool (ASP) from which they were saved or to the system ASP (ASP number 1) or to a basic user ASP (ASP numbers 2 through 32).

Some objects cannot be restored to user ASPs. More information about object types which can be restored to user ASPs is in the Backup and Recovery book, SC41-5304. If the library exists in, or is being restored to the system ASP, journals, journal receivers, and save files can be restored to basic user ASPs. All other object types will be restored to the ASP of the library.

**ATTENTION: System or product libraries (libraries that begin with a Q or #) must not be created in or restored to a user ASP. Doing so can cause unpredictable results.**

### \*SAVASP

The objects are restored to the ASP from which they were saved.

**1-32** Specifies the ASP number. When the specified ASP is 1, the specified objects are restored to the system ASP, and when the specified ASP is 2 through 32, the objects are restored to the basic user ASP specified.

---

## File to receive output (OUTFILE)

Specifies the database file to which the information is directed when \*OUTFILE is specified for the **Output (OUTPUT)** parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QASRRSTO in QSYS with the format name QSRRST as a model.

### Qualifier 1: File to receive output

*name* Specify the name of the database file to which output from the command is directed. If this file does not exist, it is created in the specified library.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

### **\*CURLIB**

The current library for the thread is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

*name* Specify the name of the library to be searched.

---

## Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when \*OUTFILE is specified for the **Output (OUTPUT)** parameter.

### Element 1: Member to receive output

#### \*FIRST

The first member in the file receives the output. If OUTMBR(\*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

*name* Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

### Element 2: Replace or add records

#### \*REPLACE

The existing records in the specified database file member are replaced by the new records.

**\*ADD** The new records are added to the existing information in the specified database file member.

Top

---

## Type of output information (INFTYPE)

Specifies the type of information which is printed or directed to the database file.

**\*OBJ** The list contains an entry for each object requested to be restored.

**\*MBR** The list contains an entry for each object, database file member, and spooled file requested to be restored.

Top

---

## Libraries to omit (OMITLIB)

Specifies a list of libraries to be excluded from the restore operation.

### Single values

#### \*NONE

No libraries are excluded from the restore operation.

### Other values (up to 300 repetitions)

#### *generic-name*

Specify the generic name of the libraries to be excluded.

**Note:** A generic name is specified as a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

*library-name*

Specify the name of the library to be excluded from the restore operation.

Top

---

## Objects to omit (OMITOBJ)

Specifies the objects to be excluded from the operation. Up to 300 objects or generic object values can be specified.

### Element 1: Object

#### Qualifier 1: Object

\*NONE

No objects are excluded from the operation.

\*ALL All objects of the specified object type are excluded from the operation.

*generic-name*

Specify the generic name of the objects to be excluded.

**Note:** A generic name is specified as a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

*name* Specify the name of the object to be excluded from the operation.

#### Qualifier 2: Library

\*ALL The specified objects are excluded from all libraries that are part of the operation.

*generic-name*

Specify the generic name of the libraries that contain objects to be excluded.

**Note:** A generic name is specified as a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

*name* Specify the name of the library that contains the object to be excluded from the operation.

### Element 2: Object type

\*ALL All object types are excluded from the operation, depending on the value specified for the object name.

*character-value*

Specify the object type of the objects to be excluded from the operation.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Top

---

## Examples

### Example 1: Restoring Most Recently Saved Version

```
RSTOBJ OBJ(PAYROLL) SAVLIB(LIBX) DEV(TAP01)
        OBJTYPE(*PGM) VOL(*SAVVOL)
```

This command restores to LIBX the program named PAYROLL that was saved from LIBX. The tape drive named TAP01 is used to restore the most recently saved version of the program.

### Example 2: Specifying Save Date and Time

```
RSTOBJ OBJ(PAY*) SAVLIB(LIBX) DEV(OPT01) VOL(ABCD)
        OPTION(*OLD) SAVDATE(102202)
        SAVTIME(143000) RSTLIB(LIBY)
```

All objects whose names start with PAY and that were saved from the library named LIBX on optical volume ABCD at 14:30:00 on 10/22/02 are restored to LIBY. Volume ABCD must be put on the optical device named OPT01. Because OPTION(\*OLD) is specified, the only objects restored are those having the same object name and type both in LIBY on the system and in LIBX on the optical volume.

### Example 3: Adding a New Program to the QGPL Library

```
RSTOBJ OBJ(NEWPROG) SAVLIB(QGPL) DEV(OPT01) OBJTYPE(*PGM)
        VOL(PGMS) OPTION(*NEW) ALWOBJDIF(*AUTL)
```

A new program named NEWPROG is added to the general purpose library, QGPL. It is restored from a volume labeled PGMS that is inserted in the optical device named OPT01. If the object was saved on a different system and was linked to an authorization list, it will be linked to the authorization list with the same name on this system.

### Example 4: Printing An Output List

```
RSTOBJ OBJ(*ALL) SAVLIB(LIB) DEV(TAP01) OBJTYPE(*PGM)
        VOL(SVOL) SEQNBR(2) SAVDATE(082392)
        SAVTIME(143000) OUTPUT(*PRINT)
```

All programs that were saved from the library named LIB that exist in the saved version on the tape volume named SVOL, sequence number 2, saved at 14:30:00 on 08/23/92, are restored to the library named LIB. An output list of all objects restored as well as objects not restored is provided.

### Example 5: Restoring Journal Receivers

```
RSTOBJ OBJ(*ALL) SAVLIB(BACKUP) DEV(*SAVF)
        OBJTYPE(*JRNRV) SAVF(SAVEJ) RSTASP(3)
```

All journal receivers that were saved from the library named BACKUP into the save file named SAVEJ are restored to the library named BACKUP. The journal receivers are restored to basic user ASP 3 (unless they already exist in the library named BACKUP and are in a different ASP or ASP 3 contains a library).

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF370C

Not authorized to ALWOBJDIF parameter.

#### CPF3705

&2 &1 in &3 not journaled.



**CPF3706**  
&2 &1 not restored to library &3.

**CPF3707**  
Save file &1 in &2 contains no data.

**CPF3709**  
Tape devices do not support same densities.

**CPF3727**  
Duplicate device &1 specified on device name list.

**CPF3728**  
Device &1 specified with other devices.

**CPF3730**  
Not authorized to &2 &1 in library &3.

**CPF3731**  
Cannot use &2 &1 in library &3.

**CPF3733**  
&2 &1 in &3 previously damaged.

**CPF3738**  
Device &1 used for save or restore is damaged.

**CPF3739**  
Database file &1 member in &3 damaged.

**CPF374C**  
No objects restored to ASP &2.

**CPF3743**  
File cannot be restored, displayed, or listed.

**CPF3748**  
Object information for library &1 damaged.

**CPF3767**  
Device &1 not found.

**CPF3768**  
Device &1 not valid for command.

**CPF3769**  
File found on media not save/restore file.

**CPF3770**  
No objects saved or restored for library &1.

**CPF3773**  
&1 objects restored. &2 not restored to &4.

**CPF3780**  
Specified file for library &1 not found.

**CPF3781**  
Library &1 not found.

**CPF3782**  
File &1 in &2 not a save file.

**CPF3783**  
Cannot determine VOL(\*SAVVOL) location. No objects restored.

**CPF3784**  
Restore device specified in the DEV parameter does not match VOL(\*SAVVOL) device.

**CPF3791**  
While processing &2 &1 in &3, encountered end of file &4.

**CPF3793**  
Machine or ASP storage limit reached.

**CPF3794**  
Save or restore operation ended unsuccessfully.

**CPF3796**  
Storage limit exceeded for user profile &4.

**CPF37A5**  
RSTASP &1 is not online.

**CPF3805**  
Objects from save file &1 in &2 not restored.

**CPF3807**  
Data decompression error for save file &1 in &2.

**CPF3812**  
Save file &1 in &2 in use.

**CPF384F**  
&2 &1 not restored to library &3.

**CPF3867**  
Contents of FILEMBR parameter not correct.

**CPF3868**  
FILEMBR specified but OBJTYPE must be \*ALL or \*FILE.

**CPF3871**  
No objects saved or restored; &3 objects not included.

**CPF3872**  
&1 objects restored. &2 not restored to &4.

**CPF38A2**  
ASP device &1 not correct.

**CPF38A6**  
Only one library allowed with specified parameters.

**CPF5729**  
Not able to allocate object &1.

**CPF9809**  
Library &1 cannot be accessed.

**CPF9812**  
File &1 in library &2 not found.

**CPF9814**  
Device &1 not found.

**CPF9820**  
Not authorized to use library &1.

**CPF9822**  
Not authorized to file &1 in library &2.

**CPF9825**

Not authorized to device &1.

**CPF9829**

Auxiliary storage pool &1 not found.

**CPFB8ED**

Device description &1 not correct for operation.

**OPT1498**

Volume name list exhausted on device &1.

**OPT1502**

Attempted to process past the end of a multi-volume set.

**OPT1605**

Media or device error occurred.

[Top](#)



---

## Restore S/36 File (RSTS36F)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Restore System/36 File (RSTS36F) command restores to the system a single file or a group of files from a save all set. A single file can be restored from a database physical file, a diskette file or a tape file. A group of files can be restored from a save all set on diskette or tape.

The restore operation reads the description of the file from the database physical file, diskette, or tape, creates a database physical or logical file and copies any data from the database physical file, diskette file or tape file into the new database file. The database physical file must have been created with the Save System/36 File (SAVS36F) command. The diskette or tape file may have been created on a System/36 using the SAVE system operator control language (OCL) procedure (or the equivalent OCL use of the \$COPY SSP utility), or by using the Save System/36 File (SAVS36F) command. The Restore System/36 File (RSTS36F) command accepts diskette or tape files created on a System/34 or System/32 using the \$COPY utility.

The RSTS36F command accepts a diskette file created as a compressed file.

If the file being restored does not exist in the library specified on the TOLIB parameter, it is created. A physical file member is added using the name syntax 'Myymmdd', which identifies the original creation date of the file. This naming convention is needed by the System/36 environment in order to support date-differentiated files.

If a file name is specified for the TOFILE parameter, the name must meet the AS/400 system naming standards. For more information about AS/400 system naming conventions, see Chapter 2, "Control Language Syntax", in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.

If TOFILE(\*SET) is specified, the files that are restored may have names that contain characters not allowed in an AS/400 simple object name. In this case, the file name is changed to an AS/400 system extended name and the file is restored.

If the name contains a blank, a single quotation mark, a double quotation mark, an asterisk, a question mark, or a device control character (hexadecimal 00 through 3F or hexadecimal FF), the invalid characters are replaced with underlines. The file is then restored using the resulting simple or extended name; for example, A\_? would become "A\_"). If a file already exists with this new name, it is handled like any other name (see the MBROPT parameter).

If a file name is changed because of invalid characters, an informational message (CPF 2C1F) is sent to the recursion level above the program that is running this command. If the name is changed from a simple name to an extended name, no message is sent.

If the restore function creates the file and the file was not previously secured, the new file is owned by the user issuing the RSTS36F command and the file is created with a default authority of \*ALL (that is the same as AUT(\*ALL)).

If the file was saved from the S/36 where the attributes were an extend value of zero or no value specified, then a default value of 32 767 divided by the record length is assigned. If an extend value of

zero is required, use the change physical file (CHGPF) command (after the restore is completed) to set SIZE(\*EXTEND) to zero. If the file was saved from the AS/400 system, the file is restored and the extend value does not change.

**Note:** This function is intended only for exchanging files with a System/36.

**Restrictions:**

1. This command is shipped with public \*EXCLUDE authority.
2. The following authorities are required when running on a system using resource security:
  - \*USE, \*SECADM, and \*ALLOBJ authorities for this command
  - \*USE authority for the library specified in the TOLIB parameter
  - \*CHANGE authority for the library specified in the TOLIB parameter when restoring a file that does not already exist on the system
  - \*CHANGE and \*OBJMGMNT authority for the existing file (needed to add a new member) when restoring a date-differentiated physical file and a file by the same name but with a different creation date
  - \*ALL authority for the file when restoring to an existing physical file with the same creation date and MBROPT(\*REPLACE) specified
  - \*CHANGE authority for the based-on physical file (this physical file was referred to as the *parent file* on System/36) if the file being restored is a System/36 alternative index file (that is, a logical file)
  - \*USE authority for the diskette device description object and \*USE authority for device file QSYSDKT in library QSYS, when restoring from diskette
  - \*USE authority for the tape device description object and \*USE authority for device file QSYSTAP in library QSYS, when restoring from tape
  - \*USE authority for the file and \*USE authority for the library that contains the file (PHYFILE parameter) if restoring from a database physical file
  - If the file doesn't exist on the system but a file authorization holder object by the same name does exist, \*ALL authority or ownership for the authorization holder object
3. There is no replace function supported when restoring System/36 alternative index files (logical files). If restoring an alternative index file, no file object by the same name can already exist in the specified library.
4. If restoring a logical file, the based-on physical file must already exist in the library specified for the TOLIB parameter.
5. The AS/400 system files that are the same as the System/36 date-differentiated files are multiple-member physical files. Date-differentiated alternative index files are not supported. The data for a physical file is stored in a member that is named using the syntax 'Myymmdd' where 'yymmdd' represents the original creation date (in year/month/day format) of the file.

Because all members of a physical file share the same file attributes (for example, record length and key information), date-differentiated files with the same name that are restored to the same library are required to have the same file attributes. If an attribute mismatch is present, the files are not restored.
6. Object-level and record-level functions, including read operations, should not be used for a file being restored by the RSTS36F command. If another operation is being done at the same time on the file (for example, moving the file or reading or adding records), the restore operation stops if it cannot allocate the file.

Top

## Parameters

Keyword	Description	Choices	Notes
TOFILE	To file	<i>Name</i> , *SET	Required, Positional 1
TOLIB	To library	<i>Name</i>	Required, Positional 2
DEV	Device	Single values: *PHYFILE Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 3
SET	Set identifier	<i>Character value</i> , # <u>SAVE</u>	Optional
IGCDTA	User specified DBCS data	* <u>NO</u> , *YES	Optional
FROMLABEL	File label	<i>Character value</i>	Optional
CRTDATE	Creation date	<i>Date</i> , * <u>NONE</u>	Optional
SEQNBR	Sequence number	1-9999, * <u>SEARCH</u>	Optional
VOL	Volume identifier	Values (up to 50 repetitions): <i>Character value</i> , * <u>MOUNTED</u>	Optional
ENDOPT	End of tape option	* <u>REWIND</u> , *LEAVE, *UNLOAD	Optional
PHYFILE	Physical file	<i>Qualified object name</i>	Optional
	Qualifier 1: Physical file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , * <u>LIBL</u> , *CURLIB	
MBROPT	Data base member option	* <u>NOREPLACE</u> , *REPLACE	Optional
DATDIFF	Date differentiated file	* <u>NO</u> , *YES	Optional

Top

### To file (TOFILE)

Specifies the name given to a single file when it is restored to the system or to a group of files from a save all set that are restored to the system. If a single file is restored, this parameter allows the file to be renamed at the time it is restored to the system.

This is a required parameter.

**\*SET** Specifies that a group of files from a save all set on diskette or tape are restored to the system.

*file-name*

Specifies the file name that is given to a single file when the file is restored to the system.

Top

### To library (TOLIB)

Specifies which library should contain the file being restored.

This is a required parameter.

Top

---

## Device (DEV)

Specifies the name of a diskette unit, the names of one or more tape devices, or an indication that the file to be restored is in a database physical file. A maximum of four tape device names can be specified. If more than one tape device is used, enter the names of the devices in the order in which they are used. Each device name must be already known on the system by a device description.

### \*PHYFILE

The file to be restored is in a database physical file. The name of the database physical file is specified by the **Physical file** prompt (PHYFILE parameter).

### *device-name*

Specify the name of the diskette unit or the names of one or more tape devices that are used for the restore operation. If more than one tape device is used, enter the names of the devices in the order in which they are used. A maximum of four tape device names can be specified.

Top

---

## Set identifier (SET)

Specifies the name used to identify the save all set files saved on the diskette or tape by the SAVE procedure or the \$COPY utility on the System/36, System/34, or System/32.

### #SAVE

The files are restored from a save all set with a set identifier of #SAVE.

### *set-identifier*

Specifies the set-identifier of the save all set. The set-identifier can be from 1-8 characters long. The first character must be alphabetic (A through Z, #, \$, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas(,), apostrophes('), and blanks.

Top

---

## User specified DBCS data (IGCDTA)

Specifies whether the file contains double-byte character set (DBCS) data.

**Note:** If a file already exists by the name specified on the **To file** prompt (TOFILE parameter), the double-byte character set (DBCS) capability of the existing file is not changed by the restore operation.

### \*NO

The files being restored may not contain DBCS data. If a file already exists in the specified library with the same name as the file being restored, and the file allows DBCS data, an informational message is sent, and the restore operation continues. The resulting file allows DBCS data (the file's IGC attributes are not changed by the restore operation).

### \*YES

The files being restored may contain DBCS data. If a file already exists in the specified library with the same name as the file being restored, and the file does not allow DBCS data, an informational message is sent, and the restore operation continues. The resulting file will not allow DBCS data (the file's IGC attributes are not changed by the restore operation).

Top



---

## File label (FROMLABEL)

For a single file restore operation, this parameter can be used to specify the label of the diskette or tape file that contains the file that is to be restored to the system. If no value is specified, the file name specified for the **To file** prompt (TOFILE parameter) is used as the diskette or tape file label.

For a group restore operation, this parameter can be used to specify the diskette or tape file label within a save all set where the restore operation is to begin. If no value is specified, the restore operation begins with the first file in the set.

If a label is specified, it must be a maximum of eight characters long.

Top

---

## Creation date (CRTDATE)

Specifies the creation date of the diskette file or tape file used for the restore operation. The specified date is changed to Julian format (cyydd) for tape or international format (yyymmdd) for diskette.

Top

---

## Sequence number (SEQNBR)

Specifies, when tape is used, which sequence number is used for the restore operation.

### \*SEARCH

The volume in the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the object is restored. If the last operation on the device specified \*LEAVE for the **End of tape option (ENDOPT)** parameter, indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If \*LEAVE was not used for the **End of tape option (ENDOPT)** parameter of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

### *1-16777215*

Specify the sequence number of the file to be used for the restore operation.

Top

---

## Volume identifier (VOL)

Specifies the volume identifiers of the tapes or diskettes used for restoring the file.

### \*MOUNTED

The volume currently placed in the device is used.

### *volume-identifier*

Specify the volume identifiers of the tapes or diskettes used for restoring the file. A maximum of 50 volume identifiers can be specified.

Top

---

## End of tape option (ENDOPT)

Specifies, only when tape is used, what positioning operation is automatically performed on the tape volume after the restore operation ends. This parameter applies only to the last reel used.

**\*REWIND**

The tape is rewound, but not unloaded.

**\*LEAVE**

The tape is not rewound.

**\*UNLOAD**

The tape is automatically rewound and unloaded after the operation ends.

Top

---

**Physical file (PHYFILE)**

Specifies the name of the database physical file that is used as the input file for the restore process. If the specified file does not exist, or is not a physical file, or the file contains no members, a message is sent. If the file contains multiple members, the first member of the file is used.

The possible library values are:

**\*LIBL** The library list is used to locate the file.

**\*CURLIB**

The current library for the job is used to locate the file. If no library is specified as the current library for the job, QGPL is used.

***library-name***

Specify the library where the file is located.

Top

---

**Data base member option (MBROPT)**

Specifies whether the data of an existing physical file member is replaced.

**\*NOREPLACE**

Data in an existing physical file member with the same name is *not* replaced, and an error message is sent to the user.

**\*REPLACE**

Data in an existing physical file member with the same name is replaced.

Top

---

**Date differentiated file (DATDIFF)**

Specifies whether the restore operation allows multiple files with the same name but different file creation dates.

**\*NO** Multiple files with the same name but different file creation dates are not allowed.

If the restore operation is done with \*NO specified, and a file already exists in the specified library with the same name, the following actions are taken:

- If the MBROPT parameter specifies the member should be replaced (\*REPLACE), the data in the member is replaced with the data from the saved file.
- If the MBROPT parameter specifies the member should not be replaced (\*NOREPLACE), a message is sent and the file is not restored.
- If no file exists by the name specified on the TOFILE parameter in the specified library, the file is restored normally.

**\*YES** Multiple files with the same name but different file creation dates are allowed.

If the restore operation is done with **\*YES** specified, and a file already exists in the specified library with the same name, the following actions are taken:

- If a member does not exist with the name of 'Myymmdd', where 'yymmdd' is the creation date of the saved file, a new member is added to the file and the data from the saved file is copied to it.
- If a member does exist with the name of 'Myymmdd', where 'yymmdd' is the creation date of the saved file, and the MBROPT parameter specifies the member should be replaced (**\*REPLACE**), the data in the member is replaced with the data from the saved file.
- If a member does exist with the name of 'Myymmdd', where 'yymmdd' is the creation date of the saved file, and the MBROPT parameter specifies the member should not be replaced (**\*NOREPLACE**), a message is sent and the file is not restored.

If a file does not already exist in the specified library with the same name, a new file is created, a member is added to the file and the data from the saved file is copied into the new member.

Top

---

## Examples

### Example 1: Restoring From Diskette

```
RSTS36F TOFILE(ACCTRCV) TOLIB(QS36F) DEV(I1)
        CRTDATE('01/22/85') VOL(SAVE1)
```

This command restores the file ACCTRCV into library QS36F. Assuming that I1 is the name of a diskette device description object, the file is restored from the diskette placed in the diskette device. The diskette must have a volume name of SAVE1. The diskette file used for the restore must have a file label of ACCTRCV and a creation date of January 22, 1985 (assuming the job date format is **\*MDY** and the date separator is a '/').

### Example 2: Restoring From Tape

```
RSTS36F TOFILE(PAY.VIEW) TOLIB(PAYLIB) DEV(T1) FROMLABEL('P*V')
        ENDOPT(*LEAVE)
```

The file P\*V is restored from device T1 as a file named PAY.VIEW in library PAYLIB. Assuming T1 is a tape device, the file is copied from one or more tapes that are on device T1. No check is made on the tape volume name. When the restore operation ends, the tape is left positioned at the end of tape file P\*V.

### Example 3: Restoring from a Physical File

```
RSTS36F TOFILE(ACCTPAY) TOLIB(QS36F) DEV(*PHYFILE)
        PHYFILE(NETLIB/SENDFILE)
```

This command restores the file ACCTPAY in library QS36F from physical file SENDFILE in library NETLIB.

### Example 4: Specifying Sequence Numbers

```
RSTS36F TOFILE(*SET) TOLIB(QS36F) DEV(T1 T2)
        SET(PAYFILES) FROMLABEL(FILE10)
        MBROPT(*REPLACE) DATDIFF(*YES)
        SEQNBR(*SEARCH) VOL(*MOUNTED) ENDOPT(*REWIND)
```

This command restores a subset of the files in the save all set called PAYFILES to library QS36F from tape. The restore operation begins with a tape file whose label is file 10. If one of the files being restored already exists in library QS36F with the same creation date as the saved file, the file is replaced. If a file already exists in library QS36F with a different creation date, a new date-differentiated number is added

to the file. The restore operation uses the tape volumes that are placed in tape drives T1 and T2. After the restore operation is complete, the last tape volume is rewound to the beginning of the tape.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2C4A

Device &1 not correct for command.

#### CPF2C4B

Duplicate device &1 specified in device name list.

#### CPF2C4C

Diskette device &1 included in multiple device specification.

#### CPF2C4D

Not all files were restored.

#### CPF2C4E

Restore operation ended before all files were restored.

#### CPF2C45

Input file &1 cannot be processed by RSTS36F.

#### CPF2C47

Existing file &1 or member &3 in library &2 not replaced.

#### CPF2C48

Input file &1 in &2 not correct for command.

#### CPF2C49

Output file &1 in &2 not correct for command.

#### CPF2C5A

Alternate index file &1 in library &2 not replaced.

#### CPF2C5E

Input file &1 in &2 not correct for command.

#### CPF2C50

File description for file &1 is not available.

#### CPF2C52

Error occurred during attempt to create file &1 in library &2.

#### CPF2C53

Member &3 not added to file &1 in library &2 because error occurred.

#### CPF9810

Library &1 not found.

#### CPF9812

File &1 in library &2 not found.

#### CPF9814

Device &1 not found.

#### CPF9820

Not authorized to use library &1.

#### CPF9822

Not authorized to file &1 in library &2.

**CPF9825**

Not authorized to device &1.

**CPF9826**

Cannot allocate file &2.

**CPF9830**

Cannot assign library &1.

**CPF9831**

Cannot assign device &1.

**CPF9845**

Error occurred while opening file &1.

**CPF9847**

Error occurred while closing file &1 in library &2.

**CPF9848**

Cannot open file &1 in library &2 member &3.

**CPF9849**

Error while processing file &1 in library &2 member &3.

**\*STATUS Messages****CPI2C11**

Copying records to file &1 in library &2 member &3.

Top



---

## Restore S/36 Library Members (RSTS36LIBM)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Restore System/36 Library Members (RSTS36LIBM) command reads a file containing library members, creates database source or data file members on this system, and copies the member data from the file into each restored member.

The input file can be a diskette file, tape file, or database physical file on this system.

The file could have been created on a System/36, a System/34, or a System/32 using either the FROMLIBR or SAVELIBR system operator control language (OCL) procedure (or the equivalent use of \$MAINT), or by using the Save System/36 Library Members (SAVS36LIBM) command.

Diskette files created on a System/34 using the BACKUP procedure or \$BACK utility are not accepted by this command. Compressed SAVELIBR diskette files (used by IBM to distribute system libraries for System/36 after release 5.0) are not accepted by this command.

In System/36 terms, the input file format could be a SAVELIBR diskette file or tape file, a record-mode LIBRFILE diskette file, tape file, or physical file, or a sector-mode LIBRFILE diskette file, tape file, or physical file. In other words, the input file can be any diskette file, tape file, or physical file created by the System/36 \$MAINT SSP utility.

If the library identified by the value specified for the **To library** prompt (TOLIB parameter) does not exist, it is created. Also, the source files QS36SRC and QS36PRC are created if they do not exist in the restore-to library. If the restore operation creates the library, the new library is owned by the user running the RSTS36LIBM command and the library is created with a default authority of \*ALL (that is, the same as AUT(\*ALL)).

If a sector-mode FROMLIBR file or a SAVELIBR file created on a System/36 is being restored, data files QS36LOD and QS36SBR may also be created to hold restored load and subroutine members. Restored load and subroutine members are not converted on the current system.

### Restrictions:

- The following authorities are required when running on a system using resource security:
  1. \*SECADM and \*ALLOBJ authorities
  2. \*USE authority for this command and \*USE authority for the Create Library (CRTLIB) command if the library needs to be created
  3. \*USE authority for the CRTSRCPF command if QS36SRC or QS36PRC must be created
  4. \*USE authority for the CRTPF command if QS36LOD or QS36SBR must be created
  5. \*CHANGE authority for the library specified in the TOLIB parameter
  6. \*CHANGE and \*OBJMGMT authority for file QS36SRC in the specified library if restoring source library members
  7. \*CHANGE and \*OBJMGMT authority for file QS36PRC in the specified library if restoring procedure library members
  8. \*CHANGE and \*OBJMGMT authority for file QS36LOD in the specified library if restoring load library members
  9. \*CHANGE and \*OBJMGMT authority for file QS36SBR in the specified library if restoring subroutine library members

10. \*USE authority for the diskette device description object, \*USE authority for device file QSYSDKT in library QSYS if restoring from diskette
  11. \*USE authority for the tape device description object and \*USE authority for device file QSYSTAP in library QSYS, if restoring from tape
  12. \*USE authority for the file and \*USE authority for the library that contains the file (PHYFILE parameter) if restoring from a database physical file
- No object-level or record-level operations should be active for files QS36SRC, QS36PRC, QS36SBR, and QS36LOD while members are being restored by RSTS36LIBM. If the necessary files cannot be allocated exclusively, no members are restored.
  - The member name or generic member name specified (TOMBR parameter) must meet AS/400 system naming standards.

If a generic member name or \*ALL is specified, a member may be selected to be restored that has a name containing characters not allowed in an AS/400 system *simple* object name. In this case, the member name is restored using the AS/400 system *extended* name syntax (for example, A!B would become "A!B").

If the name contains a blank, a single quotation mark (') a double quotation mark ("), an asterisk (\*), a question mark (?), or a device control character (hexadecimal '00'-'3F' or hexadecimal 'FF'), these characters are replaced by underlines and the member is restored using the resulting simple or extended name (for example, A\*/? would become A\_/\_ and A? would become A\_).

An informational message is sent each time invalid characters are replaced to get a valid name. An additional informational message is sent if the resulting name change caused a member to be replaced. No message is sent if a member is restored using the extended name syntax without replacing invalid characters.

Top

## Parameters

Keyword	Description	Choices	Notes
TOMBR	To member	<i>Generic name, name, *ALL</i>	Required, Positional 1
TOLIB	To library	<i>Name</i>	Required, Positional 2
DEV	Device	Single values: *PHYFILE Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 3
SRCMBRS	S/36 source members	<u>*ALL</u> , *SRC, *PRC, *NONE	Optional
MBROPT	Data base member option	<u>*NEW</u> , *OLD, *ALL	Optional
IGCDTA	User specified DBCS data	<u>*NO</u> , *YES	Optional
FROMLABEL	File label	<i>Character value</i>	Optional
CRTDATE	Creation date	<i>Date</i> , <u>*NONE</u>	Optional
SEQNBR	Sequence number	1-9999, <u>*SEARCH</u>	Optional
VOL	Volume identifier	Values (up to 50 repetitions): <i>Character value</i> , <u>*MOUNTED</u>	Optional
ENDOPT	End of tape option	<u>*REWIND</u> , *LEAVE, *UNLOAD	Optional
PHYFILE	Physical file	<i>Qualified object name</i>	Optional
	Qualifier 1: Physical file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
OBJMBRS	S/36 object members	<u>*NONE</u> , *SBR, *LOD, *ALL	Optional

Top



---

## To member (TOMBR)

Specifies the names of the members to restore.

This is a required parameter.

**\*ALL** All members of the specified member types are restored.

*member-name*

The members having the specified member name are restored.

*generic\*-member-name*

All members that have the specified generic member name are restored.

Top

---

## To library (TOLIB)

Specifies the library to contain the members to restore from the input file.

This is a required parameter.

Top

---

## Device (DEV)

Specifies the names of the devices to use for the restore operation. A maximum of four device names may be specified.

This is a required parameter.

**\*PHYFILE**

The input file is the database physical file specified by the **Physical file** prompt (PHYFILE parameter).

*device-name*

Specify the name of the diskette unit or the names of one or more tape devices that are to be used for the restore operation. If more than one tape device is used, specify the names of the devices in the order in which they are used. A maximum of four tape device names can be specified.

Top

---

## S/36 source members (SRCMBRS)

Specifies which source member types (source and procedure members on System/36) are restored.

**\*ALL** Restore all source and procedure members in the input file that match the member name specified on the **To member** prompt (TOMBR parameter).

**\*SRC** Restore only System/36 source members (to file QS36SRC) that match the member name specified on the **To member** prompt (TOMBR parameter).

**\*PRC** Restore only System/36 OCL procedure members (to file QS36PRC) that match the member name specified on the **To member** prompt (TOMBR parameter).

**\*NONE**

Do not restore any System/36 source or procedure library members.

Top

---

## Data base member option (MBROPT)

Specifies, for database files currently on the system, which file members are restored.

**\*NEW** Only new members (members that don't already exist in the appropriate file) are restored.

**\*OLD** Only old members (members that already exist in the appropriate file) are restored. The existing member is replaced by the copy of the member restored from the file.

**\*ALL** All members are restored. Members that don't already exist are created, and members that do exist are replaced.

Top

---

## User specified DBCS data (IGCDTA)

Specifies whether the source and procedure members being restored can contain double-byte character set (DBCS) data. This attribute is used if the restore operation needs to create source files QS36SRC and QS36PRC to hold the restored library members.

**Note:** If the QS36SRC or QS36PRC source file already exists in the library specified on the **To library** prompt (TOLIB parameter) and the file's DBCS capability does not match this parameter, an error message is sent and no members are restored.

**\*NO** The source or procedure members being restored cannot contain double-byte character set data.

**\*YES** The source or procedure members being restored can contain double-byte character set data.

Top

---

## File label (FROMLABEL)

Specifies the label value (eight characters maximum) of the diskette or tape file that contains the members to be restored. If \*PHYFILE is not specified for the **Device** prompt (DEV parameter), a value must be specified here.

Top

---

## Creation date (CRTDATE)

Specifies the creation date of the diskette file or tape file used for the restore operation. The specified date is changed to Julian format (cyyddd) for tape or international format (yyymmdd) for diskette.

Top

---

## Sequence number (SEQNBR)

Specifies, when tape is used, which sequence number is used for the restore operation.

### **\*SEARCH**

The volume in the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the object is restored. If the last operation on the device specified \*LEAVE for the **End of tape option (ENDOPT)** parameter, indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If \*LEAVE was not used for the **End of tape option (ENDOPT)** parameter of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

#### **1-16777215**

Specify the sequence number of the file to be used for the restore operation.

Top

---

## **Volume identifier (VOL)**

Specifies the volume identifiers of the tapes or diskettes used for restoring the file.

### **\*MOUNTED**

The volume currently placed in the device is used.

#### ***volume-identifier***

Specify the volume identifiers of the tapes or diskettes used for restoring the file. A maximum of 50 volume identifiers can be specified.

Top

---

## **End of tape option (ENDOPT)**

Specifies, only when tape is used, what positioning operation is automatically performed on the tape volume after the restore operation ends. This parameter applies only to the last reel used.

### **\*REWIND**

The tape is rewound, but not unloaded.

### **\*LEAVE**

The tape is not rewound.

### **\*UNLOAD**

The tape is automatically rewound and unloaded after the operation ends.

Top

---

## **Physical file (PHYFILE)**

Specifies the name of the database physical file that is used as the input file for the restore operation. If the specified file does not exist or is not a physical file, no library members are restored. If the file contains multiple members, the first member of the file is used.

The possible library values are:

**\*LIBL** The library list is used to locate the file.

### **\*CURLIB**

The current library for the job is used to locate the file. If no library is specified as the current library for the job, QGPL is used.

#### ***library-name***

Specify the library where the file is located.

---

## S/36 object members (OBJMBRS)

Specifies which object member types (load and subroutine members on System/36) are restored. Because the System/36 and the AS/400 system are not object compatible, any restored members are not immediately useable after the restore operation. IBM-supplied commands or user-written operations must be run to convert the object member to a useable AS/400 system object.

### \*NONE

System/36 load or subroutine library members are not restored.

**\*SBR** Only System/36 subroutine members (to file QS36SBR) that match the member name specified on the **To member** prompt (TOMBR parameter) are restored.

**\*LOD** Only System/36 load members (to file QS36LOD) that match the member name specified on the **To member** prompt (TOMBR parameter) are restored.

**\*ALL** All load and subroutine members in the input file that match the member name specified on the **To member** prompt (TOMBR parameter) are restored.

Top

---

## Examples

### Example 1: Restoring All Members

```
RSTS36LIBM  TOMBR(XYZ1)  TOLIB(JOHNSON)  DEV(I1)
             SRCMBRS(*PRC)  MBROPT(*ALL)  FROMLABEL('XYZ1')
```

This command restores the single OCL procedure member XYZ1 as a member of source file QS36PRC in library JOHNSON. Assuming I1 refers to a diskette device, the input diskette file must have the label XYZ1.

### Example 2: Restoring Members from a File

```
RSTS36LIBM  TOMBR(X*)  TOLIB(ORDER)  DEV(*PHYFILE)
             PHYFILE(NETLIB/S36SRC)
```

This command restores all source and procedure members with names starting with the character 'X' and that do not already exist as members of QS36SRC and QS36PRC in library ORDER. The members are restored from file S36SRC in library NETLIB.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2C4A

Device &1 not correct for command.

#### CPF2C4B

Duplicate device &1 specified in device name list.

#### CPF2C4C

Diskette device &1 included in multiple device specification.

#### CPF2C41

&2 members restored, &3 members not restored, &4 members excluded.

**CPF2C42**  
No members restored to library &1, &4 members excluded.

**CPF2C46**  
Input file &1 cannot be processed by RSTS36LIBM.

**CPF2C48**  
Input file &1 in &2 not correct for command.

**CPF2C49**  
Output file &1 in &2 not correct for command.

**CPF2C5E**  
Input file &1 in &2 not correct for command.

**CPF2C50**  
File description for file &1 is not available.

**CPF2C52**  
Error occurred during attempt to create file &1 in library &2.

**CPF2C53**  
Member &3 not added to file &1 in library &2 because error occurred.

**CPF2C57**  
DBCS attribute not same as existing file &1 in &2.

**CPF2C70**  
Input file &1 contains incorrect control statement record.

**CPF2C71**  
NAME keyword on COPY statement missing or incorrect.

**CPF2C72**  
LIBRARY keyword on COPY statement missing or incorrect.

**CPF9807**  
One or more libraries in library list deleted.

**CPF9808**  
Cannot allocate one or more libraries on library list.

**CPF9810**  
Library &1 not found.

**CPF9812**  
File &1 in library &2 not found.

**CPF9814**  
Device &1 not found.

**CPF9820**  
Not authorized to use library &1.

**CPF9822**  
Not authorized to file &1 in library &2.

**CPF9824**  
Not authorized to command &1 in library &2.

**CPF9825**  
Not authorized to device &1.

**CPF9826**  
Cannot allocate file &2.

**CPF9830**

Cannot assign library &1.

**CPF9845**

Error occurred while opening file &1.

**CPF9847**

Error occurred while closing file &1 in library &2.

**CPF9849**

Error while processing file &1 in library &2 member &3.

**\*STATUS Messages****CPI2C12**

Copying file &1 in library &2 label &3 to work file.

Top

---

## 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