Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 621.
Contents

Merge TCP/IP Host Table (MRGTCPHT)  1
Parameters ................................................. 1
From file (FROMFILE) ................................. 1
From member (FROMMBR) .............................. 2
File format (FILEFMT) ................................. 2
Replace host table (REPLACE) ......................... 2
Examples .................................................. 2
Error messages ........................................... 3

Work with TCP/IP Network Sts (NETSTAT).  .... 5
Parameters ................................................ 5
Option (OPTION) ........................................... 5
Examples .................................................. 5
Error messages ........................................... 6

Start DNS Query (NSLOOKUP) ....................... 7
Parameters ................................................ 10
Host (HOSTNAME) ......................................... 11
Domain Name Server (DMNNAMSVR) .................. 11
Examples .................................................. 11
Error messages ........................................... 12

Open Data Base File (OPNDBF) .................... 13
Parameters ................................................ 13
File (FILE) .................................................. 14
Open option (OPTION) ..................................... 14
Member to be opened (MBR) ............................. 14
Open file identifier (OPNID) ........................... 14
Access path to use (ACCPTH) ........................... 15
Limit to sequential only (SEQONLY) .................. 15
Commitment control active (COMMIT) ............... 15
Open scope (OPNSCOPE) ................................ 16
Duplicate key check (DUPKEYCHK) .................... 16
Type of open (TYPE) ...................................... 16
Examples .................................................. 16
Error messages ........................................... 17

Open Query File (OPNQRYF) ......................... 19
Parameters ................................................ 20
File specifications (FILE) ............................... 22
Open options (OPTION) .................................... 23
Format specifications (FORMAT) ....................... 23
Query selection expression (QRYSLT) ............... 24
Key field specifications (KEYFLD) ..................... 25
Unique key fields (UNIQUEKEY) ....................... 26
Join field specifications (JFLD) ....................... 27
Join with default values (JDFVAL) .................... 29
Join file order (JORDER) ............................... 30
Grouping field names (GRPFLD) ....................... 30
Group selection expression (GRPSLTL) .............. 31
Mapped field specifications (MAPFLD) ............... 31
Ignore decimal data errors (IGNDECCERR) .......... 34
Open file identifier (OPNID) ........................... 35
Limit to sequential only (SEQONLY) ................. 35
Commitment control active (COMMIT) ............... 35
Open scope (OPNSCOPE) ................................. 36
Duplicate key check (DUPKEYCHK) .................... 36
Allow copy of data (ALWCPYDTA) ...................... 36
Performance optimization (OPTIMIZE) ............... 37
Optimize all access paths (OPTALLAP) ............... 38
Sort sequence (SORTSEQ) ............................... 38
Language ID (LANGID) ................................... 39
Final output CCSID (CCSID) ............................ 39
Type of open (TYPE) ...................................... 39
Examples .................................................. 40
Error messages ........................................... 43

Otherwise (OTHERWISE) ......................... 47
Parameters ................................................ 47
Command (CMD) ............................................ 47
Examples .................................................. 48
Error messages ........................................... 48

Override with Data Base File (OVRDBF) ..... 49
Parameters ................................................ 49
File being overridden (FILE) ......................... 50
Overriding to data base file (TOFILE) ............. 51
Overriding member (MBR) ............................... 51
Starting position in file (POSITION) ............... 52
Record format lock (RCDFMTLCK) ..................... 53
Records to force a write (FRCRATIO) ............... 54
Rcd format selector program (FMTSLR) ............... 54
Maximum file wait time (WAITFILE) .................. 55
Maximum record wait time (WAITRCD) ............... 55
Records retrieved at once (NBRKCDM) ............... 56
EOF retry delay in sec (EOFDTLY) .................... 56
Record format level check (LEVLCCH) ............... 57
Check expiration date (EXPCCHK) ..................... 57
Inhibit write (INHWRT) .................................. 57
Secure from other overrides (SECURE) ............... 57
Override scope (OVRSCOPE) ............................ 57
Share open data path (SHARE) ......................... 58
Open scope (OPNSCOPE) ................................ 58
Limit to sequential only (SEQONLY) ................. 58
Distributed Data (DSTDTA) ............................. 60
Examples .................................................. 60
Error messages ........................................... 61

Override with Display File (OVRDSPF) . 63
Parameters ................................................ 63
File being overridden (FILE) ......................... 64
Overriding to display file (TOFILE) ............... 64
Device (DEV) .............................................. 65
Character identifier (CHRID) .......................... 65
Decimal format (DECFMT) ............................... 66
SFLEND text (SFLENDDXT) ............................ 66
User specified DBCS data (IGCDCSTA) ............... 66
DBCS extension characters (IGCENXCHR) .......... 67

© Copyright IBM Corp. 1998, 2006
Spool the data (SPOOL) .................................. 124
Output queue (OUTQ) .................................. 124
Form type (FORMTYPE) ................................. 124
Copies (COPIES) ........................................ 124
Page range to print (PAGERANGE) .................... 125
Max spooled output records (MAXRCD) ................ 125
File separators (FILESEP) .............................. 125
Spooled output schedule (SCHEDULE) ................ 125
Hold spooled file (HOLD) ................................ 126
Save spooled file (SAVE) ............................... 126
Output priority (on OUTQ) (OUTPTY) ................. 126
User data (USRDATA) ................................... 126
Spool file owner (SPLFOWN) ............................ 127
User Defined Option (USRDFNOPT) ...................... 127
User Defined Data (USRFNDATA) ....................... 128
User Defined Object (USRFNOBJ) ....................... 128
Spool file name (SPLFNAME) ............................ 129
Expiration date for file (EXPDATE) .................... 129
Days until file expires (DAYS) .......................... 130
User specified DBCS data (IGCDFTA) .................... 130
DBCS extension characters (IGCEXNCHR) .............. 130
DBCS character rotation (IGCCRRKT) .................... 130
DBCS characters per inch (IGCCPI) .................... 131
DBCS SO/SI spacing (IGCSSI) .......................... 131
DBCS coded font (IGCCDFNT) ........................... 131
Maximum file wait time (WAITFILE) ................... 132
Record format level check (LVLCHK) .................... 132
Secure from other overrides (SECURE) ................. 133
Override scope (OVRSCOPE) ........................... 133
Share open data path (SHARE) ......................... 133
Open scope (OPNSCOPE) ................................ 133
Examples .............................................. 134
Error messages ......................................... 134

Override with Save File (OVRSAVF) .......................... 135
Parameters ............................................. 135
File being overridden (FILE) .......................... 135
Save file (TOFILE) ...................................... 136
Extend file (EXTEND) .................................... 136
Starting position in file (POSITION) .................... 137
Maximum file wait time (WAITFILE) .................... 137
Secure from other overrides (SECURE) ................. 137
Override scope (OVRSCOPE) ........................... 138
Share open data path (SHARE) ......................... 138
Open scope (OPNSCOPE) ................................ 138
Examples .............................................. 138
Error messages ......................................... 138

Override with Tape File (OVRTAPF) .......................... 141
Parameters ............................................. 141
File being overridden (FILE) .......................... 142
Overriding to tape file (TOFILE) ....................... 143
Device identifier (DEV) .................................. 143
Volume identifier (VOL) .................................. 143
Tape reels specifications (REELS) ...................... 144
Sequence number (SEQNBR) ............................. 146
File label (LABEL) ...................................... 146
Record length (RCDLLEN) .............................. 146
Block length (BLKLEN) .................................. 147
Buffer offset (BUFOFSET) .............................. 147
Record block format (RCDBLKFM) ....................... 148
Extend file (EXTEND) .................................... 149
Tape density (DENSITY) .................................. 150
Data compaction (COMPACT) ............................ 154
Code (CODE) ............................................ 154
Creation date (CRDTDATE) ............................. 155
File expiration date (EXPDATE) ......................... 155
End of tape option (ENDOPT) ........................... 155
User label program (USRLBLPGM) ....................... 156
Conversion table (TBL) .................................. 156
From CCSID (FROMCCSID) ............................... 157
To CCSID (TOCCSID) .................................... 157
User specified DBCS data (IGCDFTA) .................... 157
Maximum file wait time (WAITFILE) .................... 157
Secure from other overrides (SECURE) ................. 158
Override scope (OVRScope) ............................ 158
Share open data path (SHARE) ......................... 158
Open scope (OPNSCOPE) ................................ 158
Examples .............................................. 159
Error messages ......................................... 159

Parameter Definition (PARM) .......................... 161
Parameters ............................................. 161
Keyword (KWD) ......................................... 163
Type of value (TYPE) .................................... 163
Value length (LEN) ..................................... 165
Return value (RTNVAL) .................................. 166
Constant value (CONSTANT) ............................ 166
Restricted values (RSTD) ............................... 167
Default value (DFT) ..................................... 167
Valid values (VALUES) ................................. 168
Relational expression (REL) ............................. 169
Range of values (RANGE) ............................... 169
Special values (SPCVAL) ............................... 170
Single values (SNGVAL) ............................... 170
Minimum values required (MIN) ......................... 171
Maximum values allowed (MAX) ......................... 172
Allow unprintable characters (ALWUNPRT) ............ 172
Allow variable names (ALWVAR) ......................... 172
Is PARM a program name (PGM) ......................... 173
Is PARM a data area name (DTAARA) ................. 173
If a file parameter, how used (FILE) .................. 173
Full field required (FULL) ............................. 174
Value an expression (EXPR) ............................ 174
Varying length (VARY) ................................. 174
Pass attribute byte (PASSATR) ......................... 175
Value to pass if unspecified (PASSVAL) .............. 175
Case of value (CASE) .................................... 176
CCSID of value (CCSID) ............................... 176
List displacement (LISTDISPL) ......................... 176
Display input (DSPINPUT) ............................. 176
Choice text (CHOICE) .................................. 177
Choice program (CHOICEPGM) ........................... 177
Prompt control (PMTCTL) .............................. 178
Prompt control program (PMTCLPGM) .................... 178
Key parameter (KEYPARM) ............................. 179
Initial prompt length (INLPMTLEN) ..................... 179
Prompt specifications (PROMPT) ......................... 179
Examples .............................................. 180
Error messages ......................................... 181
<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
<th>Topics</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>183</td>
<td>Program (PGM)</td>
<td>Parameters, Parameter CL variable names (PARM), Examples, Error messages</td>
<td></td>
</tr>
<tr>
<td>185</td>
<td>Verify TCP/IP Connection (PING)</td>
<td>Parameters, Remote system (RMTSYS), Remote internet address (INTNETADR), Address version format (ADRFVFMT), Message mode (MSGMODE), Packet length (PKTLEN), Number of packets (NBRPKT), Wait time (WAITTIME), Local internet address (LCLINTNTRA), Type of service (TOS), IP time to live (IPTTL), Examples, Error messages</td>
<td></td>
</tr>
<tr>
<td>191</td>
<td>PM iSeries Line Control (PMLINMON)</td>
<td>Parameters, Examples, Error messages</td>
<td></td>
</tr>
<tr>
<td>193</td>
<td>Prompt Control Definition (PMTCTL)</td>
<td>Parameters, Controlling keyword (CTL), Controlling conditions (COND), Number of true conditions (NBRTRUE), Logical relation (LGLREL), Examples, Error messages</td>
<td></td>
</tr>
<tr>
<td>197</td>
<td>Position Data Base File (POSDBF)</td>
<td>Parameters, Open file identifier (OPNID), File position (POSITION), Examples, Error messages</td>
<td></td>
</tr>
<tr>
<td>199</td>
<td>Print Adopting Objects (PRTADPOBJ)</td>
<td>Parameters, User profile (USRPRF), Changed report only (CHGRPTONLY), Examples, Error messages</td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>Print AFP Data (PRTAFPDTA)</td>
<td>Parameters, File (FILE), Member (MBR), Print device (DEV), Form definition (FORMMD), Number of copies (COPIES), Starting page (STRPAGE), Ending page (ENDPAGE), Print fidelity (FIDELITY), Examples, Error messages</td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>Print Command Usage (PRTCMDUSG)</td>
<td>Parameters, Command (CMD), Program (PGM), Examples, Error messages</td>
<td></td>
</tr>
<tr>
<td>209</td>
<td>Print Communications Security (PRTCMNSEC)</td>
<td>Parameters, Changed report only (CHGRPTONLY), Examples, Error messages</td>
<td></td>
</tr>
<tr>
<td>213</td>
<td>Print Communications Trace (PRTCMNTRC)</td>
<td>Parameters, Configuration object (CFGOBJ), From stream file (FROMSTMF), Type (CFGTYPE), Output (OUTPUT), File to receive output (OUTFILE), Output member options (OUTMBR), Character code (CODE), Line description (SLTLIND), Controller description (SLTCP), Format SNA data only (FMTSNA), Format RR, RNR commands (FMTTRR), Format TCP/IP data (FMTTCP), Format LCP data (FMTLCP), Format NCP data (FMTNCP), Format TCP/IP data by address (TCPIPADDR), IP port number (SLPORT), Format UI data only (FMTUI), Format MAC or SMT data only (FMTMAC), Format Ethernet data only (FMTETH), Format call control data (FMTCCD), Format broadcast data (FMTBCD), Exclude LMI data (EXCLM), Format LMI data only (FMTLMI), Format HPR over IP data only (FMTHPRI), Format LDLC over IP data only (FMTLDC), Examples, Error messages</td>
<td></td>
</tr>
<tr>
<td>223</td>
<td>Print Device Addresses (PRTDEVADR)</td>
<td>Parameters, Controller description (CTLD), Examples, Error messages</td>
<td></td>
</tr>
<tr>
<td>225</td>
<td>Print Directory Information (PRTDIRINF)</td>
<td>Parameters, Type of report (RPTTYPE), Information file prefix (INFFILEPFX)</td>
<td></td>
</tr>
</tbody>
</table>
Information library (INFLIB) .............................. 226
Owners (OWNER) ........................................... 226
Objects (OBJ) ............................................ 227
Object types (OBJTYPE) ................................. 227
Directory subtree (SUBTREE) ............................ 227
Smallest size (MINSIZE) .................................. 228
Sort by (SORT) ............................................ 228
Include path names (INCPATH) ......................... 228
Examples .................................................. 228
Error messages .......................................... 229

Print Document (PRTDOC) ............................... 231
Parameters ................................................ 231
Document (DOC) .......................................... 233
Folder (FLR) ............................................. 233
Display print options (OPTIONS) ....................... 233
Printer file (PRTFILE) .................................... 234
File to receive output (OUTFILE) ....................... 235
Output member options (OUTMBR) ...................... 235
Current status value (CURSTS) ......................... 236
New status value (NEWSTS) ............................. 236
Type of data for output (OUTDATATYPE) ............. 236
Delete document (DLTDTC) ............................. 236
Output device (OUTPUT) ............................... 237
Print device (DEV) ....................................... 237
Output queue (OUTQ) .................................... 237
Output file (SPLFILE) .................................... 238
Form type (FORMTYPE) ................................... 238
Print separator page (COVERPAGE) ..................... 238
Print quality (PRTQUALITY) ............................ 238
Number of copies (COPIES) ............................ 239
Print on both sides (DUPLEX) .......................... 239
Automatic page binding (AUTOBIND) ................. 239
Delay printing (HOLD) ................................... 240
Print document error log (PRTERRLOG) ............... 240
Error log form type (ERRFORM) ....................... 240
Large print (LARGEPRINT) ............................. 240
Merge type (MRTYPE) .................................... 241
Query (QRYQDF) ......................................... 241
Data document (DTADOC) .............................. 241
Data folder (DTAFLD) ................................. 241
Data file (DTAFILE) ..................................... 242
Data member (DTAMBR) .................................. 242
Multiple line report (MLTLINEPRT) ................. 242
Adjust line endings (ADJLINES) ....................... 243
Adjust page endings (ADJPAGES) ....................... 243
Allow widow lines (ALWWIDOW) ....................... 243
Renumber system page numbers (RENUMBER) ....... 243
Print change symbols (PRTCHGSYM) .................. 244
Change change symbols to print (SYMBOLS) ........ 244
Draft spacing (DRAFTSPACE) .......................... 244
Print line numbers (LINNRBR) ......................... 244
Resolve instructions (RESOLVE) ....................... 245
Additional spaces to left (LEFTSPACES) ............. 245
Character identifier (CHRID) .......................... 245
Save resolved output (SAVOUTPUT) ................. 245
Resolved output document (SAVDOC) ................. 246
Resolved output folder (SAVFILR) .................... 246
Place on job queue (JOBQ) ............................ 246
Job description (JOBID) .............................. 246
Send completion message (SNDMMSG) ................. 247
Cancel on error (CNLERR) ............................ 247
Start page (STRPAGE) .................................. 247
End page (ENDPAGE) .................................... 248
Page ranges (PAGRANGE) ............................... 248
Number of labels across page (LBLACROSS) ....... 249
Width of labels (LBLLWIDTH) ......................... 249
Sheet feed labels (SHEETFEED) ...................... 249
Number of rows per sheet (LBLSHDOWN) .......... 249
Shift left margin (SHFLTLMR) ......................... 249
Examples ................................................ 250
Error messages .......................................... 251

Print Disk Information (PRTDSKINF) ............... 255
Parameters ................................................ 255
Type of report (RTYPE) .................................. 256
ASP device (ASPDEV) .................................... 256
Libraries (LIB) .......................................... 256
Owners (OWNER) ......................................... 256
Folders (FLR) ............................................ 257
Documents (DOC) ........................................ 257
Objects (OBJ) ............................................ 257
Object types (OBJTYPE) ............................... 257
Smallest size (MINSIZE) ............................... 258
Sort by (SORT) .......................................... 258
Examples ................................................ 258
Error messages .......................................... 259

Print Error Log (PRTERRLOG) ......................... 261
Parameters ................................................ 261
Type of log data to list (TYPE) ......................... 262
Logical device (DEV) .................................... 262
Resource name (RSRCNAME) ........................... 263
Error log identifier (ERLOGID) ....................... 263
Output (OUTPUT) ........................................ 263
Time period for log output (PERIOD) ............... 263
Print format (PRTFMT parameter) ..................... 264
File to receive output (OUTFILE) ..................... 265
Output member options (OUTMBR) ..................... 265
Volume type (VOLTYPE) ............................... 266
Model number (MODEL) ............................... 266
Volume (VOL) ........................................... 266
Volume statistical data (VOLSTAT) .................... 266
Volume statistics type (VOLSTATTYPE) ............... 267
Error log entries to select (SELECT) ............... 267
Sort by (SORT) .......................................... 267
Examples ................................................ 268
Error messages .......................................... 268

Print Internal Data (PRTINTDATA) .................. 271
Parameters ................................................ 271
Type of data (TYPE) ..................................... 272
Dump identifier (DMPID) ............................... 272
Time period for internal data (PERIOD) ............. 272
Job name (JOB) ......................................... 274
Thread ID to include (SLITHD) ....................... 274
Server type (SVRTYPE) ............................... 274
Examples ................................................ 275
Error messages .......................................... 275

Print IP over SNA (PRTIPSCFG) ....................... 277
Parameters ........................................ 277
Examples ............................................ 277
Error messages ....................................... 277

Print JOBD Authority (PRTJOBDAUT) 279
Parameters ............................................ 280
Library (LIB) ......................................... 280
Changed report only (CHGRPTONLY) ............... 281
Examples .............................................. 281
Error messages ...................................... 281

Print Profile Internals (PRTPRFINT) 283
Parameters ............................................ 284
Select by (SELECT) ..................................... 284
User profile (USRPRF) ................................ 285
Percent full (PCTFULL) ............................... 285
Examples .............................................. 285
Error messages ...................................... 285

Print Publicly Auth Objects
(PRTPUBAUT) .......................................... 287
Parameters ............................................ 288
Object type (OBJTYPE) ............................... 289
Changed report only (CHGRPTONLY) ............... 289
Library (LIB) ......................................... 289
Print file authority (FILAUTH) ...................... 290
Print command authority (CMDAUT) ............... 290
Print program authority (PGMAUT) ............... 291
Print JOBD authority (JOBDAUT) .................. 291
Folder (FLR) .......................................... 291
Directory (DIR) ....................................... 291
Search subdirectory (SCHSUBDIR) .................. 292
Examples .............................................. 292
Error messages ...................................... 292

Print Private Authorities (PRTPVTAUT) 293
Parameters ............................................ 295
Object type (OBJTYPE) ............................... 295
Changed report only (CHGRPTONLY) ............... 295
Library (LIB) ......................................... 296
Authority type (AUTTYPE) ......................... 296
Folder (FLR) .......................................... 296
Print AUTL objects (AUTLOBJ) ..................... 296
Directory (DIR) ....................................... 297
Search subdirectory (SCHSUBDIR) .................. 297
Examples .............................................. 297
Error messages ...................................... 297

Print Queue Authority (PRTQAUT) 299
Parameters ............................................ 299
Library (LIB) ......................................... 300
Changed report only (CHGRPTONLY) ............... 301
Examples .............................................. 301
Error messages ...................................... 301

Print Subsystem Description
(PRTSSSDAUT) ........................................ 303
Parameters ............................................ 304
Library (LIB) ......................................... 304
Changed report only (CHGRPTONLY) ............... 305
Examples .............................................. 305
Error messages ...................................... 305

Print SQL Information (PRTSQINF) 307
Parameters ............................................ 307
Object (OBJ) .......................................... 307
Object type (OBJTYPE) ............................... 308
Examples .............................................. 308
Error messages ...................................... 308

Print Stop Word List (PRTSWL) 309
Parameters ............................................ 309
Language ID (LANGID) ............................... 309
Stop word list type (TYPE) ......................... 309
Examples .............................................. 309
Error messages ...................................... 309

Print System Information
(PRTSYSINF) ........................................ 311
Parameters ............................................ 311
Examples .............................................. 311
Error messages ...................................... 311

Print System Security Attr
(PRTSYSSECA) ........................................ 313
Parameters ............................................ 313
Examples .............................................. 313
Error messages ...................................... 313

Print Point-to-Point Profile
(PRTTCPPTP) ........................................ 315
Parameters ............................................ 315
Configuration profile (CFGPRF) ..................... 315
Examples .............................................. 315
Error messages ...................................... 315

Print Trace Data (PRTTRC) ........................................ 317
Parameters ............................................ 317
Data member (DTAMBR) ............................... 318
Data library (DTALIB) ............................... 318
Select jobs (SLTJOB) ................................. 318
Delete trace (DLITRRC) .............................. 319
Sort by (SORT) ........................................ 319
Output (OUTPUT) ..................................... 319
File to receive output (OUTFILE) ................. 320
Output member options (OUTMBR) ................. 320
Examples .............................................. 320
Error messages ...................................... 321

Print Trigger Programs (PRTRGPGM) 323
Parameters ............................................ 323
Library (LIB) ......................................... 323
Changed report only (CHGRPTONLY) ............... 324
Examples .............................................. 324
Error messages ...................................... 325

Print User Objects (PRTUSROBJ) 327
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Option (OPTION)</th>
<th>Library (LIB)</th>
<th>Examples</th>
<th>Error messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>391</td>
<td>391</td>
<td>392</td>
<td>392</td>
<td>393</td>
</tr>
</tbody>
</table>

**Reclaim DDM Conversations**

*(RCLDDMCNV)* | 395 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>395</td>
</tr>
<tr>
<td>Examples</td>
<td>395</td>
</tr>
<tr>
<td>Error messages</td>
<td>395</td>
</tr>
</tbody>
</table>

**Reclaim Document Lib Object**

*(RCLDLO)* | 397 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>397</td>
</tr>
<tr>
<td>Document library object (DLO)</td>
<td>398</td>
</tr>
<tr>
<td>In folder (FLR)</td>
<td>398</td>
</tr>
<tr>
<td>System object name (SYSOBJNAM)</td>
<td>399</td>
</tr>
<tr>
<td>System object attributes (SYSOBJATTR)</td>
<td>399</td>
</tr>
<tr>
<td>Auxiliary storage pool ID (ASP)</td>
<td>399</td>
</tr>
<tr>
<td>File to receive output (OUTFILE)</td>
<td>399</td>
</tr>
<tr>
<td>Output member options (OUTMBR)</td>
<td>400</td>
</tr>
<tr>
<td>Examples</td>
<td>400</td>
</tr>
<tr>
<td>Error messages</td>
<td>401</td>
</tr>
</tbody>
</table>

**Reclaim Library (RCLLIB)** | 403 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>403</td>
</tr>
<tr>
<td>Library (LIB)</td>
<td>403</td>
</tr>
<tr>
<td>Examples</td>
<td>403</td>
</tr>
<tr>
<td>Error messages</td>
<td>403</td>
</tr>
</tbody>
</table>

**Reclaim Object Links (RCLLNK)** | 405 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>406</td>
</tr>
<tr>
<td>Object (OBJ)</td>
<td>406</td>
</tr>
<tr>
<td>Directory subtree (SUBTREE)</td>
<td>406</td>
</tr>
<tr>
<td>Damaged object option (DMGOBJOPT)</td>
<td>407</td>
</tr>
<tr>
<td>Examples</td>
<td>407</td>
</tr>
<tr>
<td>Error messages</td>
<td>408</td>
</tr>
</tbody>
</table>

**Reclaim Objects by Owner**

*(RCLOBJOWN)* | 409 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>410</td>
</tr>
<tr>
<td>User profile (USRPRF)</td>
<td>410</td>
</tr>
<tr>
<td>Examples</td>
<td>410</td>
</tr>
<tr>
<td>Error messages</td>
<td>410</td>
</tr>
</tbody>
</table>

**Reclaim Optical (RCLOPT)** | 411 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>411</td>
</tr>
<tr>
<td>Optical media library (MLB)</td>
<td>411</td>
</tr>
<tr>
<td>Option (OPTION)</td>
<td>411</td>
</tr>
<tr>
<td>Volume identifier (VOL)</td>
<td>412</td>
</tr>
<tr>
<td>Rebuild directory index (DIR)</td>
<td>412</td>
</tr>
<tr>
<td>Examples</td>
<td>412</td>
</tr>
<tr>
<td>Error messages</td>
<td>413</td>
</tr>
</tbody>
</table>

**Reclaim Resources (RCLRSC)** | 415 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>416</td>
</tr>
<tr>
<td>Program level (LVL)</td>
<td>416</td>
</tr>
<tr>
<td>Close option (OPTION)</td>
<td>416</td>
</tr>
</tbody>
</table>

**Reclaim Spool Storage (RCLSPLSTG)** | 421 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>421</td>
</tr>
<tr>
<td>Days (DAYS)</td>
<td>421</td>
</tr>
<tr>
<td>Examples</td>
<td>422</td>
</tr>
<tr>
<td>Error messages</td>
<td>422</td>
</tr>
</tbody>
</table>

**Reclaim Storage (RCLSTG)** | 423 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>424</td>
</tr>
<tr>
<td>Estimate time required (ESTIMATE)</td>
<td>424</td>
</tr>
<tr>
<td>Select (SELECT)</td>
<td>425</td>
</tr>
<tr>
<td>Omit (OMIT)</td>
<td>425</td>
</tr>
<tr>
<td>ASP device or group (ASPDEV)</td>
<td>425</td>
</tr>
<tr>
<td>Examples</td>
<td>425</td>
</tr>
<tr>
<td>Error messages</td>
<td>427</td>
</tr>
</tbody>
</table>

**Reclaim Temporary Storage**

*(RCLTMPSTG)* | 429 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>429</td>
</tr>
<tr>
<td>Library (LIB)</td>
<td>430</td>
</tr>
<tr>
<td>Days unused (DAYS)</td>
<td>430</td>
</tr>
<tr>
<td>Examples</td>
<td>431</td>
</tr>
<tr>
<td>Error messages</td>
<td>431</td>
</tr>
</tbody>
</table>

**Receive Distribution (RCVDST)** | 433 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>433</td>
</tr>
<tr>
<td>Distribution identifier (DSTID)</td>
<td>434</td>
</tr>
<tr>
<td>User identifier (USRID)</td>
<td>434</td>
</tr>
<tr>
<td>Document (DOC)</td>
<td>434</td>
</tr>
<tr>
<td>In folder (FLR)</td>
<td>435</td>
</tr>
<tr>
<td>File to receive output (OUTFILE)</td>
<td>435</td>
</tr>
<tr>
<td>Output member options (OUTMBR)</td>
<td>435</td>
</tr>
<tr>
<td>Type of data for output (OUTDTATYP)</td>
<td>436</td>
</tr>
<tr>
<td>Acknowledge receipt (ACKRCV)</td>
<td>437</td>
</tr>
<tr>
<td>Distribution ID extension (DSTIDEXN)</td>
<td>437</td>
</tr>
<tr>
<td>Keep in mail log (KEEP)</td>
<td>438</td>
</tr>
<tr>
<td>Command character identifier (CMDCHRID)</td>
<td>438</td>
</tr>
<tr>
<td>Examples</td>
<td>438</td>
</tr>
<tr>
<td>Error messages</td>
<td>439</td>
</tr>
</tbody>
</table>

**Receive File (RCVF)** | 441 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>441</td>
</tr>
<tr>
<td>Display device (DEV)</td>
<td>442</td>
</tr>
<tr>
<td>Record format (RCDFMT)</td>
<td>442</td>
</tr>
<tr>
<td>Open file identifier (OPNID)</td>
<td>442</td>
</tr>
<tr>
<td>Wait (WAIT)</td>
<td>442</td>
</tr>
<tr>
<td>Examples</td>
<td>443</td>
</tr>
<tr>
<td>Error messages</td>
<td>443</td>
</tr>
</tbody>
</table>

**Receive Journal Entry (RCVJRNE)** | 445 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>446</td>
</tr>
<tr>
<td>Journal (JRNL)</td>
<td>448</td>
</tr>
<tr>
<td>Program to receive entries (EXITPGM)</td>
<td>448</td>
</tr>
<tr>
<td>Journaled physical file (FILE)</td>
<td>451</td>
</tr>
<tr>
<td>Objects (OBJ)</td>
<td>452</td>
</tr>
<tr>
<td>Objects (OBJPATH)</td>
<td>454</td>
</tr>
<tr>
<td>File identifier (OBJFID)</td>
<td>455</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>Lock state (LOCK)</td>
<td>536</td>
</tr>
<tr>
<td>Source update options (SRCOPT)</td>
<td>537</td>
</tr>
<tr>
<td>Source sequence numbering (SRCSEQ)</td>
<td>537</td>
</tr>
<tr>
<td>Record format (RCDFMNT)</td>
<td>538</td>
</tr>
<tr>
<td>Examples.</td>
<td>538</td>
</tr>
<tr>
<td>Error messages.</td>
<td>538</td>
</tr>
<tr>
<td><strong>Release Communications Device</strong> (RLSCMNDEV)</td>
<td>541</td>
</tr>
<tr>
<td>Parameters</td>
<td>541</td>
</tr>
<tr>
<td>Device (DEV)</td>
<td>541</td>
</tr>
<tr>
<td>Examples.</td>
<td>542</td>
</tr>
<tr>
<td>Error messages.</td>
<td>542</td>
</tr>
<tr>
<td><strong>Release Distribution Queue</strong> (RLSDSTQ)</td>
<td>543</td>
</tr>
<tr>
<td>Parameters</td>
<td>543</td>
</tr>
<tr>
<td>Distribution queue (DSTQ)</td>
<td>543</td>
</tr>
<tr>
<td>Priority (PTY)</td>
<td>543</td>
</tr>
<tr>
<td>Examples.</td>
<td>544</td>
</tr>
<tr>
<td>Error messages.</td>
<td>544</td>
</tr>
<tr>
<td><strong>Release File System Locks</strong> (RLSIFSLCK)</td>
<td>547</td>
</tr>
<tr>
<td>Parameters</td>
<td>547</td>
</tr>
<tr>
<td>Remote location (RMTLOCNAME)</td>
<td>547</td>
</tr>
<tr>
<td>Object (OBJ)</td>
<td>548</td>
</tr>
<tr>
<td>Examples.</td>
<td>548</td>
</tr>
<tr>
<td>Error messages.</td>
<td>548</td>
</tr>
<tr>
<td><strong>Release Job</strong> (RLSJOB)</td>
<td>551</td>
</tr>
<tr>
<td>Parameters</td>
<td>551</td>
</tr>
<tr>
<td>Job name (JOB)</td>
<td>551</td>
</tr>
<tr>
<td>Duplicate job option (DUPJOBOPT)</td>
<td>552</td>
</tr>
<tr>
<td>Examples.</td>
<td>552</td>
</tr>
<tr>
<td>Error messages.</td>
<td>552</td>
</tr>
<tr>
<td><strong>Release Job Queue</strong> (RLSJBOQ)</td>
<td>555</td>
</tr>
<tr>
<td>Parameters</td>
<td>555</td>
</tr>
<tr>
<td>Job queue (JOBQ)</td>
<td>555</td>
</tr>
<tr>
<td>Examples.</td>
<td>555</td>
</tr>
<tr>
<td>Error messages.</td>
<td>556</td>
</tr>
<tr>
<td><strong>Release Job Schedule Entry</strong> (RLSJBOSCDE)</td>
<td>557</td>
</tr>
<tr>
<td>Parameters</td>
<td>557</td>
</tr>
<tr>
<td>Job name (JOB)</td>
<td>557</td>
</tr>
<tr>
<td>Entry number (ENTRYNBR)</td>
<td>558</td>
</tr>
<tr>
<td>Examples.</td>
<td>558</td>
</tr>
<tr>
<td>Error messages.</td>
<td>558</td>
</tr>
<tr>
<td><strong>Release Output Queue</strong> (RLSOUTQ)</td>
<td>561</td>
</tr>
<tr>
<td>Parameters</td>
<td>561</td>
</tr>
<tr>
<td>Output queue (OUTQ)</td>
<td>561</td>
</tr>
<tr>
<td>Examples.</td>
<td>561</td>
</tr>
<tr>
<td>Error messages.</td>
<td>562</td>
</tr>
<tr>
<td><strong>Release Reader</strong> (RLSRDR)</td>
<td>563</td>
</tr>
<tr>
<td>Parameters</td>
<td>563</td>
</tr>
<tr>
<td>Reader (RDR)</td>
<td>563</td>
</tr>
<tr>
<td>Examples.</td>
<td>563</td>
</tr>
<tr>
<td>Error messages.</td>
<td>563</td>
</tr>
<tr>
<td><strong>Release Remote Phase</strong> (RLSRMTPHS)</td>
<td>565</td>
</tr>
<tr>
<td>Parameters</td>
<td>565</td>
</tr>
<tr>
<td>Phase (PHASE)</td>
<td>566</td>
</tr>
<tr>
<td>Plan (PLAN)</td>
<td>566</td>
</tr>
<tr>
<td>Application identifier (APPID)</td>
<td>566</td>
</tr>
<tr>
<td>Release Remote Phase (RMTLOCNAME)</td>
<td>566</td>
</tr>
<tr>
<td>Device description (DEV)</td>
<td>566</td>
</tr>
<tr>
<td>Examples.</td>
<td>566</td>
</tr>
<tr>
<td>Error messages.</td>
<td>567</td>
</tr>
<tr>
<td><strong>Release Spooled File</strong> (RLSSPLF)</td>
<td>569</td>
</tr>
<tr>
<td>Parameters</td>
<td>569</td>
</tr>
<tr>
<td>Spooled file (FILE)</td>
<td>570</td>
</tr>
<tr>
<td>Job name (JOB)</td>
<td>570</td>
</tr>
<tr>
<td>Spooled file number (SPLNBR)</td>
<td>570</td>
</tr>
<tr>
<td>Job system name (JOBSYSNAME)</td>
<td>571</td>
</tr>
<tr>
<td>Spooled file created (CRTDATE)</td>
<td>571</td>
</tr>
<tr>
<td>Select files for (SELECT)</td>
<td>572</td>
</tr>
<tr>
<td>ASP device (ASpdev)</td>
<td>572</td>
</tr>
<tr>
<td>Examples.</td>
<td>573</td>
</tr>
<tr>
<td>Error messages.</td>
<td>573</td>
</tr>
<tr>
<td><strong>Release Writer</strong> (RLSWTR)</td>
<td>575</td>
</tr>
<tr>
<td>Parameters</td>
<td>575</td>
</tr>
<tr>
<td>Writer (WTR)</td>
<td>575</td>
</tr>
<tr>
<td>Resume writing at (OPTION)</td>
<td>575</td>
</tr>
<tr>
<td>Starting page (PAGE)</td>
<td>576</td>
</tr>
<tr>
<td>Examples.</td>
<td>576</td>
</tr>
<tr>
<td>Error messages.</td>
<td>576</td>
</tr>
<tr>
<td><strong>Remove Directory</strong> (RMDIR)</td>
<td>579</td>
</tr>
<tr>
<td>Parameters</td>
<td>580</td>
</tr>
<tr>
<td>Directory (DIR)</td>
<td>580</td>
</tr>
<tr>
<td>Remove link (RMVLNK)</td>
<td>580</td>
</tr>
<tr>
<td>Examples.</td>
<td>580</td>
</tr>
<tr>
<td>Error messages.</td>
<td>581</td>
</tr>
<tr>
<td><strong>Remove Access Code</strong> (RMVACC)</td>
<td>583</td>
</tr>
<tr>
<td>Parameters</td>
<td>583</td>
</tr>
<tr>
<td>Document access code (ACC)</td>
<td>583</td>
</tr>
<tr>
<td>Examples.</td>
<td>583</td>
</tr>
<tr>
<td>Error messages.</td>
<td>584</td>
</tr>
<tr>
<td><strong>Remove Autostart Job Entry</strong> (RMVAJE)</td>
<td>585</td>
</tr>
<tr>
<td>Parameters</td>
<td>585</td>
</tr>
<tr>
<td>Subsystem description (SBSD)</td>
<td>585</td>
</tr>
<tr>
<td>Job name (JOB)</td>
<td>586</td>
</tr>
<tr>
<td>Examples.</td>
<td>586</td>
</tr>
<tr>
<td>Error messages.</td>
<td>586</td>
</tr>
<tr>
<td><strong>Remove Alert Description</strong> (RMVALRD)</td>
<td>587</td>
</tr>
<tr>
<td>Parameters</td>
<td>587</td>
</tr>
<tr>
<td>Message identifier (MSGID)</td>
<td>587</td>
</tr>
</tbody>
</table>
Alert table (ALRTBL) .......................... 587
Examples .................................. 588
Error messages ............................. 588

Remove Auth List Entry (RMVAUTLE) 591
Parameters .................................. 591
Authorization list (AUTL) .................. 591
User (USER) .................................. 592
Examples .................................. 592
Error messages ............................. 592

Remove Breakpoint (RMVBKP) .............. 593
Parameters .................................. 593
Statement identifier (STMT) ............... 593
Program (PGM) ............................. 594
Examples .................................. 594
Error messages ............................. 594

Remove Binding Directory Entry
(RMVBNDDIRE) ............................. 595
Parameters ................................. 595
Binding directory (BNDDIR) ............... 595
Object specifications (OBJ) .............. 596
Examples .................................. 596
Error messages ............................. 597

Remove Cfg List Entries (RMVCFGLE) 599
Parameters .................................. 599
Configuration list type (TYPE) ............. 600
APPN local location entry (APPNLCLE) .... 600
APPN remote location entry (APPNRMTE) ... 601
Configuration list (CFGL) ................. 601
Async network address entry (ASYNCADRE) 601
Async remote location entry (ASYNCLOCCE) ... 601
Retail pass-through entry (RTLPASTHRE) ... 602
Filtered control point name (FTRCPNAME) ... 602
Filtered CP network identifier (FTRCPNETID) ... 602
Local location name (LCLLOCNAME) ........ 603
SNA pass-through entry (SNAPASTHRE) ..... 603
Examples .................................. 603
Error messages ............................. 603

Remove Cluster Node Entry
(RMVCLUNODE) ............................. 605
Parameters .................................. 606
Cluster (CLUSTER) .......................... 606
Node identifier (NODE) ................. 606
Examples .................................. 606
Error messages ............................. 606

Remove Communications Entry
(RMVCMNE) .................................. 607
Parameters .................................. 607
Subsystem description (SBSD) ............. 607
Device (DEV) ............................... 608
Remote location (RMTLOCNAME) ........... 608
Mode (MODE) .................................. 609
Examples .................................. 609
Error messages ............................. 609

Remove Community for SNMP
(RMVCOMSNMP) .............................. 611
Parameters .................................. 611
Community name (COM) .................... 611
Translate community name (ASCICOM) ..... 611
Examples .................................. 612
Error messages ............................. 612

Remove CRG Device Entry
(RMVCRGDEVE) .............................. 613
Parameters .................................. 613
Cluster (CLUSTER) .......................... 614
Cluster resource group (CRG) .............. 614
Configuration object list (CFGOBJ) ......... 614
Examples .................................. 614
Error messages ............................. 615

Remove CRG Node Entry
(RMVCRGNODE) .............................. 617
Parameters .................................. 618
Cluster (CLUSTER) .......................... 618
Cluster resource group (CRG) .............. 618
Node identifier (NODE) ........................ 618
Examples .................................. 618
Error messages ............................. 619

Appendix. Notices ............................ 621
Trademarks ................................. 622
Terms and conditions ........................ 623
Merge TCP/IP Host Table (MRGTCPHT)

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

The Merge TCP/IP Host Table (MRGTCPHT) command is used to merge host names, internet addresses, and text comment entries from a physical file member into the local host table. A replace option is also provided that allows the entire local host table to be replaced by the host table entries in a user specified physical file member.

A file format option is provided that allows either *AS400, *AIX, or *NIC file format to be merged with the local host table. The local host table is located in member QUSRYS/QATOCHOST.HOSTS and is created as a physical file.

A maximum of 4 host names per IP address is allowed when host tables are merged. For example: If the local host table already has 3 host names and the physical file member to be merged has 2 additional host names, only the first host name in the physical file is merged into the final host table. Host names that exist both in the local host table and the physical file member being merged are not duplicated.

Attention: The original copy of the local host table is not saved by the Merge TCP/IP Host Table (MRGTCPHT) command. To save the original host table, create a copy of the file QUSRYS/QATOCHOST.HOSTS using the Copy File (CPYF) command. Do this before issuing the MRGTCPHT command.

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

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROMFILE</td>
<td>From file</td>
<td>Qualified object name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: From file</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>FROMMBR</td>
<td>From member</td>
<td>Name, *FIRST, *LAST</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>FILEFMT</td>
<td>File format</td>
<td>*AS400, *AIX, *NIC</td>
<td>Optional</td>
</tr>
<tr>
<td>REPLACE</td>
<td>Replace host table</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
</tbody>
</table>

From file (FROMFILE)

Specifies the physical file that contains the member being used for the merge operation.

Qualifier 1: From file
name Specify the name of the physical file.

Qualifier 2: Library

*LIBL All libraries in the job’s library list are searched.

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

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

From member (FROMMBR)

Specifies the physical file member to be used in the merge operation.

*FIRST The first member of the physical file is used to merge with the host table.

*LAST The last member of the physical file is used to merge with the host table.

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

File format (FILEFMT)

Specifies the format of the physical file member to be merged with the local host table.

*AS400 The physical file member to be merged with the local host table is *AS400 format.

Note: *AS400 can only be used if the physical file member specified is a host table from an iSeries running Version 3, Release 1, Modification 0 (V3R1M0) or later of OS/400. If you import a host table from a system running any version of OS/400 prior to V3R1M0, specify *AIX.

*AIX The physical file member to be merged with the local host table is *AIX format.

*NIC The physical file member to be merged with the local host table is *NIC format.

Replace host table (REPLACE)

Specifies whether the physical file member is to be merged with or replaces the local host table.

*NO The physical file member is merged with the local host table.

*YES The physical file member replaces the local host table.

Examples

Example 1: Replacing Local Host Table
MRGTCPHT FROMFILE(AS400FILE) REPLACE(*YES) FILEFMT(*AS400)
This command replaces the contents of QUSRYS/QATOCHOST.HOSTS with the contents of the first member of physical file AS400FILE. The first member of physical file AS400FILE is in *AS400 host table format.

Example 2: Merging Local Host Table
MRGTCPHT FROMFILE(HOSTLIB/NICFILE) FROMMBR(NEWHOSTS) FILEFMT(*NIC)

This command merges the current contents of the local host table with the contents of the NEWHOSTS member of physical file NICFILE in library HOSTLIB. The physical file is in *NIC format. The records are converted from *NIC format to *AS400 format by this command.

Error messages

*ESCAPE Messages
TCP1927
   Records of file &1, member &2 not valid.
TCP1929
   Host table not available.
TCP1934
   Merge file &1, member &3, in library &2 not found.
TCP8050
   *IOSYSCFG authority required to use &1.
Work with TCP/IP Network Sts (NETSTAT)

Where allowed to run: Interactive environments (*INTERACT  
*IPGM  *IREXX  *EXEC)

Threadsafe: No

Use the Work with TCP/IP Network Status (WRKTCPSTS) command, also known as NETSTAT, to get information about the status of TCP/IP network routes, interfaces, TCP connections and UDP ports on your local system. You can also use NETSTAT to end TCP/IP connections and to start or end TCP/IP interfaces.

If IP over SNA (IPS) is enabled, NETSTAT displays information about the IP over SNA interfaces, routes, and connections. You can also use NETSTAT to end IP over SNA connections and to start or end IP over SNA interfaces.

To use this command, either the TCP/IP protocol stack or IP over SNA must be active. If neither is active, Netstat issues an escape message.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTION</td>
<td>Option</td>
<td>*SELECT, *IFC, *RTE, *CNN</td>
<td>Optional, Positional 1</td>
</tr>
</tbody>
</table>

### Option (OPTION)

Specifies which TCP/IP status information you want to work with.

*SELECT
  Display the Work with TCP/IP Network Status menu.

*IFC    Display the Work with TCP/IP Interface Status list.

*RTE    Display the Display TCP/IP Route Information list.

*CNN    Display the Work with TCP/IP Connection Status list.

### Examples

**Example 1: Displaying the Work with TCP/IP Network Status Menu**

```
WRKTCPSTS  
-0*-
WRKTCPSTS  OPTION(*SELECT)
```

Either of these commands will display the Work with TCP/IP Network Status menu.

**Example 2: Using the OPTION Parameter**

© Copyright IBM Corp. 1998, 2006
WRKTCPSTS  OPTION(*CNN)

This command displays the Work with TCP/IP Connection Status panel.

Example 3: Using a Positional Parameter
WRKTCPSTS   *RTE

The OPTION parameter is a positional parameter. The OPTION keyword is not required. This command starts NETSTAT, and the Display TCP/IP Route Information panel is shown.

Error messages
*ESCAPE Messages

TCP2670
Not able to complete request. TCP/IP services are not available.

TCP3844
Data for interface &3 not available.

TCP3881
Data for list not available.

TCP3882
Data not available.

TCP9999
Internal system error in program &1.
Start DNS Query (NSLOOKUP)

Where allowed to run: Interactive environments (*INTERACT
*IPCM *IREXX *EXEC)

Threadsafe: No

Start DNS Query (STRDNSQRY), and its alias NSLOOKUP, start the NSlookup (Name Server Lookup) tool.

NSLookup is an interactive query tool that allows you to retrieve information from, or test the response of a DNS server. You can verify that a DNS server is responding correctly before you configure your system to use it. You can also retrieve DNS information about hosts, domains, and DNS servers.

Note: NSLookup asks for (queries) information from DNS servers. To begin a NSLookup query session, an active DNS server must be designated the ‘default’ server for the query session. The default server is the DNS server that NSLookup sends all queries to unless you tell it otherwise. All references in the following help to ‘the default server’, or ‘the default DNS server’, refer only to the default DNS server for the current NSLookup query session.

NSLookup retrieves information from DNS servers. It needs an active DNS server to send its queries to. If you do not specify a DNS server with DMNAMSVR when you start the tool, it will attempt to set one of the following as its default DNS server for the session: 1. The DNS server your system is configured to use, or 2. The DNS server that is running on your local system.

If neither of these conditions exist, NSLookup will not be able to retrieve any information until you specify a DNS server to query. DMNAMSVR allows you to start the query session and set the DNS server of your choice as the default server for the session.

There are two parameters for this command:
1. HOSTNAME
2. DMNAMSVR

These parameters are used with STRDNSQRY to specify a default DNS server for the query session or, to request information about a specific host on session start up. Help for these parameters follows the list of session subcommands.

Following is a list of NSLookup subcommands that can be used once the query session is started.

NAME
Show the IP address of the host NAME. Substitute a host name for NAME. The current or ‘default’ DNS server is queried.

NAME1 NAME2
Show the IP address of the host NAME (NAME1), but query NAME2 for the information instead of the current (default) DNS server (where NAME2 is name of a DNS server).

Allows you to direct the query to a DNS server other than the current or ‘default’ DNS server for the query session.

help (or ?)
Displays a list of subcommands for the STRDNSQRY (NSLOOKUP) tool.

server NAME
Change the default (current) DNS server to NAME (where NAME is the name of a DNS server), using the current (default) DNS server.
lserv NAME
Change the default (current) DNS server to NAME (where NAME is the name of a DNS server), using the initial default DNS server.

Useful if you switched default DNS servers during your query session, and the current DNS server cannot resolve the new DNS server name. lserver allows you to make the switch using your initial default DNS server instead of the current one. If the initial DNS server also cannot resolve the new DNS name, substitute the IP address for the name, if you know it. If you do not know the IP address for the new DNS server, try restarting the NSLookup session using the DMNNAMSVR parameter to specify the new DNS server as the default server for the query session.

root
Makes the root DNS server the default DNS server for the query session. The root DNS server is defined by the ‘set root=NAME’ option.

set
The set subcommand allows you to set values for query session options. Valid option values for the set subcommand are:

set all
Show the current values for all of the session options. If no option values have been set, the default values for each option are shown.

set debug
Show debugging information.

set nodebug
Do not show debugging information.

set d2
Show exhaustive (verbose) debugging information.

set nod2
Do not show exhaustive (verbose) debugging information.

set defname
Append the default domain name to each query. The default domain name is defined by the ‘set domain=NAME’ option.

set nodefname
Do not append the default domain name to each query.

set search
Use the srchlist option instead of the defname option. Uses the list of domain names defined by the ‘set srchlist=N1/N2/N3...’ option.

set nosearch
Do not use the srchlist option.

set recurse
Query other DNS servers if the default server does not have the information.

set norecurse
Do not query other DNS servers if the default server does not have the information.

set vc
Use TCP for queries instead of UDP.

set novc
Do not use TCP for queries instead of UDP.

set ignoretc
Do not retry query using TCP if UDP reply is truncated.

set noignoretc
Retry query using TCP if UDP reply is truncated.
set domain=NAME
Set default domain name to NAME (substitute a domain name for NAME). Defines the
default domain name used by the ‘set defname’ option.

set srchlist=N1/N2/N3...
Creates a list of domain names to append to each query. Each domain name in the list is
appended to the query until a reply is received, or there are no more names in the list.
Substitute domain names for N1, N2, N3, etc.

set root=NAME
Set root server to NAME (substitute a DNS server name for NAME). Defines the server
used by the ‘root’ subcommand.

set retry=X
Set the number of retries to X (where X is a numerical value).
Note: The default value for number of retries is 1. The retry value works together with
the timeout value, which is the time in seconds that NSLookup waits before making the
first retry. Retry values are usually set to 1 or 2.

set timeout=X
Set initial timeout interval to X seconds (where X is a numerical value).
Note: timeout=X determines how long NSLookup waits before making the first retry if no
reply is received on the first query. The timeout value doubles after each unsuccessful
retry. The default value is 5 seconds.

set type=X
Determines the type of DNS record that the DNS server will use to answer the query.
Substitute ‘X’ for one of the following DNS record types:
A  IP Address record. This is the default value.
ANY Any record type that exists for the subject of the query.
CNAME Canonical Name record. Returns a list of aliases for the true (canonical) host
name if any exist.
HINFO Host information. Information about the CPU type and operating system of
subject of the query.
MX Mail Exchange record.
NS Name server (DNS server) information for the zone
PTR Pointer record. Returns a host name for an IP address.
SOA Start of Authority record.
TXT Text record.
WKS Well-known services or applications available on this host.
Note: This type of information record is not usually available.

set port=X
Use TCP/IP port ‘X’ to query the DNS server, where ‘X’ is a TCP/IP port number. The
default value is port 53.
Note: The well known port number for DNS servers is 53 and most DNS servers use it.
You do not normally need to set the port value unless the DNS server you want to query
is not using port 53. Other ports are sometimes used under special circumstances. To query DNS server that is not using port 53, set the port value to the same port number the DNS server is using.

**ls** List. The list subcommand is used to display information or write it to a file. It is used with additional values to determine the kind of information displayed or written, and if written, the path and file name of the file to write the information to. Values for the ls subcommand are:

**ls** **DOMAIN** > **FILE**
Write a list of IP addresses in **DOMAIN** to **FILE**. Substitute the name of the domain for **DOMAIN**, and the full path and filename to write to for **FILE**.

```bash
ls company.us.com > /temp/filename.extension
```

**ls** **-a** **DOMAIN**
List all canonical (true) names and aliases in **DOMAIN** (substitute a domain name for **DOMAIN**).

**ls** **-h** **DOMAIN**
List HINFO (CPU type and operating system) for **DOMAIN** (substitute a domain name for **DOMAIN**).

**ls** **-s** **DOMAIN**
List the well-known services available on **DOMAIN** (substitute a domain name for **DOMAIN**).

**ls** **-d** **DOMAIN**
List all available records for **DOMAIN** (substitute a domain name for **DOMAIN**). Includes all DNS record types.

**ls** **-t** **TYPE** **DOMAIN**
List all DNS **TYPE** records for **DOMAIN**. Substitute a DNS record type for **TYPE**, and a domain name for **DOMAIN**. See the ‘set type=X’ subcommand for a list of DNS record types.

**view** **FILE**
Display the contents of ls output **FILE** (substitute the ls output file name for **FILE**).

**exit** End the query session. Then hit enter to return to the command line.

---

**Parameters**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HOSTNAME</strong></td>
<td>Host</td>
<td>Character value, *NONE</td>
<td>Optional, Positional 1</td>
</tr>
<tr>
<td><strong>DMNNAMSVR</strong></td>
<td>Domain Name Server</td>
<td>Character value, *CFG</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Host (HOSTNAME)

Specify the TCP/IP host that you want the Domain Name Service (DNS) server to supply information about. You can use either the host’s worded name, or its numerical IP address. For example, a host’s name is anybiz.usa.com, and its numerical address is 123.4.56.7. Use either the name to obtain the address, or the address to obtain the name.

*NONE
No host name is provided. This is the default value for the HOSTNAME parameter and is used if you do not supply a host name or address. An interactive NSlookup session will be started.

host-name
Specify the name of the host to use for the DNS server query.

host-IP-address
Specify the IP address of the host to use for the DNS server query.

Domain Name Server (DMNNAMSVR)

Specify the name or the IP address of the DNS server that NSLookup will use as its default server for the query session.

Note: NSLookup retrieves information from DNS servers. It needs an active DNS server to send its queries to. If you do not specify a DNS server with DMNNAMSVR when you start the tool, it will attempt to set one of the following as its default DNS server for the session: 1. The DNS server your system is configured to use, or 2. The DNS server that is running on your local system.

If neither of these conditions exist, NSLookup will not be able to retrieve any information until you specify a DNS server to query. DMNNAMSVR allows you to start the query session and set the DNS server of your choice as the default server for the session.

Use the DMNNAMSVR parameter of the STRDNSQRY command to specify a default DNS server for your NSLookup query session. You can specify any DNS server your TCP/IP network has access to. Or, if you want to test the response of a DNS server prior to designating it for use by your system, specify that server.

*CFG Use the DNS server that is currently designated for use by this system.

domain-name-server-name
Specify the name of a DNS server.

domain-name-server-IP-address
Specify the IP address of a DNS server.

Examples

STRDNSQRY HOSTNAME('9.12.234.14') DMNNAMSVR(*CFG)

This command starts a DNS query using the host located at IP address 9.12.234.14 and the DNS server that is currently designated for use by this system.
Error messages

None
Open Data Base File (OPNDBF)

Where allowed to run: All environments (*ALL)

Threadsafe: Conditional

The Open Database File (OPNDBF) command opens a database file member. Processing of records is done later by application programs that do shared open operations.

Restrictions:
- This command is conditionally threadsafe. In multithreaded jobs, this command:
  - Is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA.
  - Is not threadsafe and fails for Distributed Data Management (DDM) files of type *SNA.
  - Is not threadsafe for logical files that require a format selector program.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>File</td>
<td>Qualified object name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: File</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>OPTION</td>
<td>Open option</td>
<td>*INP, *OUT, *ALL</td>
<td>Required, Positional 2</td>
</tr>
<tr>
<td>MBR</td>
<td>Member to be opened</td>
<td>*FIRST, *LAST</td>
<td>Optional, Positional 3</td>
</tr>
<tr>
<td>OPNID</td>
<td>Open file identifier</td>
<td>Name, *FILE</td>
<td>Optional, Positional 4</td>
</tr>
<tr>
<td>ACCPTH</td>
<td>Access path to use</td>
<td>*FILE, *ARRIVAL</td>
<td>Optional</td>
</tr>
<tr>
<td>SEQONLY</td>
<td>Limit to sequential only</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element list</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Sequential only</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Number of records</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>COMMIT</td>
<td>Commitment control active</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OPNSCOPE</td>
<td>Open scope</td>
<td>*ACTGRPDEF, *ACTGRP, *JOB</td>
<td>Optional</td>
</tr>
<tr>
<td>DUPKEYCHK</td>
<td>Duplicate key check</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>TYPE</td>
<td>Type of open</td>
<td>*NORMAL, *PERM</td>
<td>Optional</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 1998, 2006
**File (FILE)**

Specifies the file that contains the member to be opened. Overrides currently in effect are processed.

This is a required parameter.

**Qualifier 1: File**

name  Specify the name of file.

**Qualifier 2: Library**

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

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

name  Specify the name of the library to be searched.

---

**Open option (OPTION)**

Specifies the options to use to open a file. The options chosen on the first full open operation of a file are not changed on subsequent shared options.

This is a required parameter.

*INP  The file is opened only for input operations.

*OUT  The file is opened only for output operations.

*ALL  The file is opened for all operations (input, output, update, and delete).

---

**Member to be opened (MBR)**

Specifies the member to open in the database file.

*FIRST  The first member of the specified file is used.

*LAST  The last member created in the file is opened.

member-name  Specify the name of the member to be opened.

---

**Open file identifier (OPNID)**

Specifies the identifier used for naming this open operation so it can be referred to when the member is closed or positioned. This identifier must be specified on the Close File (CLOF) command, and on the Position Database File (POSDBF) command. It is not used on another Open Database File (OPNDBF) command until the file is closed, or an escape message is sent and the open operation fails.
**FILE**  The file name is used for the open operation identifier.

Specify the name used to identify this open operation.

---

**Access path to use (ACCPTH)**

Specifies which access path to use for this open operation.

**FILE**  The file access path is used. If the file is keyed, the keyed access path is used; otherwise, the arrival sequence path is used.

**ARRIVAL**  The arrival sequence access path is used. If the file is keyed, the keyed access path is ignored.

---

**Limit to sequential only (SEQONLY)**

Specifies, for database files whose records are normally processed in sequential order, whether sequential-only processing is used on the file. This parameter also specifies the number of records transferred as a group to or from the database file if sequential-only processing is used. If an override specifying sequential only processing is in effect, it takes precedence over what is specified on this parameter.

**Note:** If *ALL is specified for the Open option (OPTION) parameter or *YES is specified for the Commitment control active (COMMIT) parameter, the *NO value is used for this parameter.

**Element 1: Sequential only**

**NO**  The database file does not use sequential-only processing.

**YES**  The database file uses sequential-only processing.

**Element 2: Number of records**

**integer**  The file uses sequential-only processing. This parameter value indicates the number of records the database blocks up in its internal buffer before actually accessing the data in the member. Specifying this number is not required. If this value is not specified, the database chooses a default value.

---

**Commitment control active (COMMIT)**

Specifies whether this file is placed under commitment control.

Before a database file is opened under commitment control, the user must ensure that all files in the commitment transaction are journaled. If only the after images are being journaled, the system implicitly begins journaling both the before and the after images for the duration of the changes being made to files opened under this commitment definition.

**NO**  This file is not placed under commitment control.

**YES**  This file is placed under commitment control.
Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

*ACTGRPDFN

The scope of the open operation is determined by the activation group of the program that called the OPNDBF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

*ACTGRP

The scope of the open data path (ODP) is the activation group. Only those shared opens from the same activation group can share this ODP. This ODP is not reclaimed until the activation group is deactivated, or until the Close File (CLOF) command closes the activation group.

*JOB

The scope of the open operation is the job in which the open operation occurs.

Duplicate key check (DUPKEYCHK)

Specifies whether duplicate key checking is done on input and output operations opened by this command.

*NO

No duplicate key feedback is provided on input and output commands.

*YES

Duplicate key feedback is provided on input and output commands.

Type of open (TYPE)

Specifies the recursion level at which the reclaim resources function (RCLRSC) is effective.

Note: This parameter is not valid when the Open scope (OPNSCOPE) parameter is specified.

*NORMAL

Allow the reclaim resources function to close the file if the program exits without doing a close operation.

*PERM

The file remains open until a close operation is done using the Close File (CLOF) command, or until the routing step ends. The open data path (ODP) remains in existence even if the Reclaim Resources (RCLRSC) command is used.

Examples

OPNDBF FILE(MASTER/PAYROLL) OPTIONS(*INP)

This command opens the first member in the file PAYROLL for input processing. The open identifier associated with this open operation has the file name as its identifier. If the file is specified as SHARE(*YES), subsequent open operations of the file PAYROLL (such as in an application program) perform more efficiently and use the same ODP.
Error messages

*ESCAPE Messages

CPF4125
    Open of member &3 file &1 in &2 failed.

CPF4174
    OPNID(&4) for file &1 already exists.

CPF4175
    Output only and MBR(*ALL) cannot be used together.

CPF4176
    File &1 in &2 not a data base file.

CPF432A
    Open not allowed under commitment control; reason code &8.

CPF4327
    Commitment control resource limit exceeded.

CPF4328
    Member &4 not journaled to journal &6.

CPF4329
    Cannot associate journal &6 with commitment definition &9.

CPF8361
    Cannot place resource under commitment control. Reason code &1.

CPF8367
    Cannot perform commitment control operation.
Open Query File (OPNQRYF)

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

The Open Query File (OPNQRYF) command opens a file to a set of database records that satisfies a database query request. Once opened, the file looks like a database file opened using the Open Database File (OPNDBF) command, and the records in the file are accessed by high-level language programs that share the open data path (ODP). The path is closed, and all query resources are deallocated, using the Close File (CLOF) command.

This command is used to do any combination of the following database functions:
- Join records from more than one file, member, and record format. The join may be either equal or non-equal in nature.
- Calculate new field values using numeric and character operations on field values and constants.
- Group records by like values of one or more fields, and calculate aggregate functions, such as minimum field value and average field value, for each group.
- Select a subset of the available records, with selection both before and after grouping the records.
- Arrange result records by the value of one or more key fields.

Restrictions:
1. The user can use overrides to change the file, library, and member names specified for the FILE parameter. Overrides are ignored for the file and library specified for the FORMAT parameter, unless FORMAT(*FILE) is specified. Parameter values specified on an override command, other than TOFILE, MBR, LVLCHK, WAITRCD, SEQONLY, or INHWRT and SHARE, are ignored by the OPNQRYF command.
2. The OPNQRYF command does not share an existing open data path (ODP) in the job or activation group. If an existing SHARE(*YES) ODP in the job or activation group has the same file, library, and member name as the open query file open data path (ODP), the query file does not open and an escape message is sent.
3. Each subsequent shared open operation must use the same open options (such as SEQONLY) that are in effect when the OPNQRYF command is run.
4. Some system functions (such as the Display Physical File Member (DSPPFM) and Copy File (CPYF) commands) do not share an existing open data path. The OPNQRYF command cannot be used with those functions.
5. The file opened with the OPNQRYF command cannot be used in programs written in BASIC because BASIC does not share an existing open data path.
6. This command is conditionally threadsafe. In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA. This command is also not threadsafe and fails for Distributed Data Management (DDM) files of type *SNA.
7. Users of this command must have the following authorities:
   - Execute (*EXECUTE) authority for any library that is needed to locate the files specified for the FILE and FORMAT parameters
   - Object operational (*OBJOPR) authority for any physical or logical file specified for the FILE parameter, and one or more of the following data authorities for the physical file or based-on physical file members of a logical file member:
     - Read (*READ) authority if the file is opened for input (using option *INP)
     - Add (*ADD) authority if the file is opened for output (using option *OUT)
- Update (*UPD) authority if the file is opened for updates (using option *UPD)
- Delete (*DLT) authority if the file is opened for deletions (using option *DLT)
- *READ, *ADD, *UPD, and *DLT authority if the file is opened for all I/O operations (using option *ALL)
  
  • *OBJOPR authority for any file specified for the FORMAT parameter
  • Use (*USE) authority for any translate tables specified for the MAPFLD parameter (using option *USE)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FILE</strong></td>
<td>File specifications</td>
<td>Values (up to 32 repetitions): *Element list</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>Element 1: File</td>
<td>Qualified object name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 1: File</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 2: Member</td>
<td>Name, *FIRST, *LAST, *ALL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 3: Record format</td>
<td>Name, *ONLY</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FORMAT</strong></td>
<td>Format specifications</td>
<td>Single values: *FILE, *Element list</td>
<td>Optional</td>
</tr>
<tr>
<td>Element 1: File</td>
<td>Qualified object name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 1: File</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 2: Record format</td>
<td>Name, *ONLY</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>QRYSLT</strong></td>
<td>Query selection expression</td>
<td>Character value, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>KEYFLD</strong></td>
<td>Key field specifications</td>
<td>Single values: *NONE, *FILE</td>
<td>Optional</td>
</tr>
<tr>
<td>Element 1: Key field</td>
<td>Qualified object name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 1: Key field</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 2: Key field or element</td>
<td>Name, *MAPFLD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 2: Key field order</td>
<td>*ASCEND, *DESCEND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 3: Order by absolute value</td>
<td>*ABSVAL.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>UNIQUEKEY</strong></td>
<td>Unique key fields</td>
<td>1-120, *NONE, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------</td>
<td>------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>JFLD</td>
<td>Join field specifications</td>
<td>Single values: *NONE  Other values (up to 50 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: From field</td>
<td>Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: From field</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: File or element</td>
<td>Name, *MAPFLD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: To field</td>
<td>Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: To field</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: File or element</td>
<td>Name, *MAPFLD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32</td>
<td></td>
</tr>
<tr>
<td>JDFTVAL</td>
<td>Join with default values</td>
<td>*NO, *YES, *ONLYDFT</td>
<td>Optional</td>
</tr>
<tr>
<td>JORDER</td>
<td>Join file order</td>
<td>*ANY, *FILE</td>
<td>Optional</td>
</tr>
<tr>
<td>GRPFLD</td>
<td>Grouping field names</td>
<td>Single values: *NONE  Other values (up to 50 repetitions): Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Grouping field names</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: File or element</td>
<td>Name, *MAPFLD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32</td>
<td></td>
</tr>
<tr>
<td>GRPSLT</td>
<td>Group selection expression</td>
<td>Character value, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>MAPFLD</td>
<td>Mapped field specifications</td>
<td>Single values: *NONE  Other values (up to 50 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Mapped field</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Field definition expression</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 4: Length</td>
<td>0-32766</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 5: Decimal positions</td>
<td>0-63</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 6: Mapped field CCSID</td>
<td>1-65535, *CALC, *HEX</td>
<td></td>
</tr>
<tr>
<td>IGNDECERR</td>
<td>Ignore decimal data errors</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OPNID</td>
<td>Open file identifier</td>
<td>Name, *FILE</td>
<td>Optional</td>
</tr>
<tr>
<td>SEQONLY</td>
<td>Limit to sequential only</td>
<td>Single values: *NO  Other values: Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Sequential only</td>
<td>*YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Number of records</td>
<td>1-32767</td>
<td></td>
</tr>
<tr>
<td>COMMIT</td>
<td>Commitment control active</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OPNSCOPE</td>
<td>Open scope</td>
<td>*ACTGRPDFN, *ACTGRP, *JOB</td>
<td>Optional</td>
</tr>
<tr>
<td>DUPKEYCHK</td>
<td>Duplicate key check</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>ALWCYDTA</td>
<td>Allow copy of data</td>
<td>*YES, *OPTIMIZE, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------</td>
<td>---------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>OPTIMIZE</td>
<td>Performance optimization</td>
<td>Single values: *ALLIO, *MINWAIT</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Performance</td>
<td>*FIRSTIO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>optimization</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Number of records</td>
<td>1-2147483647</td>
<td></td>
</tr>
<tr>
<td>OPTALLAP</td>
<td>Optimize all access paths</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>SRTSEQ</td>
<td>Sort sequence</td>
<td>Single values: *HEX, *JOB, *LANGIDSHR,</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>*LANGIDUNQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Sort sequence</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>LANGID</td>
<td>Language ID</td>
<td>Name, *JOB</td>
<td>Optional</td>
</tr>
<tr>
<td>CCSID</td>
<td>Final output CCSID</td>
<td>1-65535, *JOB, *HEX</td>
<td>Optional</td>
</tr>
<tr>
<td>TYPE</td>
<td>Type of open</td>
<td>*NORMAL, *PERM</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**File specifications (FILE)**

Specifies one or more files, members, and record formats processed by the open query file. All files specified must be physical or logical database files, or Distributed Data Management (DDM) files. If Distributed Data Management files are used, all files they refer to must be on the same target system.

When more than one file, member, and record format is specified, the query joins field values to create a single set of records. Any file specified in the list may be a join logical file or view logical file member. More information on view logical files is in SQL Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

You can specify 32 values for this parameter.

This is a required parameter.

**Element 1: File**

**Qualifier 1: File**

`name` Specify the name of the file to be processed.

**Qualifier 2: Library**

*LIBL The library list is used to locate the database file.

*CURLIB

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

`name` Specify the name of the library where the database file is located.
Element 2: Member

**FIRST**
The oldest member created is to be used.

**LAST**
The newest member created is to be used.

**ALL**
All members of a partition file are to be used.

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

Element 3: Record format

**ONLY**
The only record format in the file is to be used. If the file has more than one record format, a record format name must be specified.

name Specify the name of the record format to be used. The record format must exist in the database file specified in the first element of this parameter.

Open options (OPTION)
Specifies the open option used for the query file. The options chosen on the first full open of a file are not changed on subsequent shared opens. You can either specify **ALL** or a value that combines **INP, OUT, UPD, and DLT** in a list of up to four values in any order.

Single values

**ALL** Open the file for all operations (**INP, OUT, UPD, DLT**).

Other values (up to 4 repetitions)

**INP** Open the file for input. **INP** is the only value allowed if join processing or group processing is requested, if UNIQUEKEY processing is specified, if all the fields in the open query file record format specified for the Format specifications (FORMAT) parameter are for input-only use, or if a temporary file is required to run the query.

**OUT** Open the file for output. In some high-level languages, output to certain files (such as files defined as 'direct access' in the high-level language program) is done by using a combination of input and update operations. **UPD** and **INP** are specified, or **ALL** is specified, in order to use an open query file with such a program.

**UPD** Open the file for update operations. If an input operation comes before an update, you must specify **INP** when **UPD** is specified.

**DLT** Open the file for delete operations. If a delete operation is preceded by an input operation, you must specify **INP** when **DLT** is specified.

Format specifications (FORMAT)
Specifies the record format used for records available through the open query file. The simple field names in the open query file record format must represent fields that are either defined on the Mapped field specifications (MAPFLD) parameter or are unique across all files, members, and record formats specified on the File specifications (FILE) parameter. The value for any field that has the same name as a field specified on the MAPFLD parameter is determined by the mapped-field-definition on the MAPFLD.
parameter. The value for any field not defined on the MAPFLD parameter is determined by a mapping of the field with the same name in one of the based-on files, members, and record formats specified for the FILE parameter. Only the name, type, length, decimal positions, keyboard shift, and usage attributes of each field specified in the record format that is identified on the Format specifications (FORMAT) parameter are used for the open query file. All other attributes are ignored. The attributes do not have to be the same. If they differ, the fields are mapped in a way similar to the Change Variable (CHGVAR) command.

Single values

*FILE The record format of the first or only entry on the File specifications (FILE) parameter is used.
*FILE is not allowed when more than one file, member, and record format are specified on the FILE parameter (requiring a join query).

Element 1: File

Qualifier 1: File

name Specify the name of a physical or logical database file, or a Distributed Data Management (DDM) file that contains the record format to be used.

Qualifier 2: Library

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

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

Element 2: Record format

*ONLY The only record format in the file is used. If no record format name is specified, *ONLY is the default. If the file has more than one record format, a record format name must be specified.

name Specify the name of the record format to be used. The record format must exist in the database file specified for the first element of this parameter.

Query selection expression (QRYSLT)

Specifies the selection values used (before grouping) to determine the records that are available through the open query file.

*ALL All records in the physical or logical files, members, and record formats specified for the File specifications (FILE) parameter (after join processing, if required) are selected.

'query-selection'

Specify an expression (contained in apostrophes) that describes the values used to determine which records are selected. You can specify any logical expression formed from relationships (such as *EQ and *NE) of field and constant values or functions of field and constant values. At
least one field name is specified in each relationship. However, you cannot specify a field that
depends on an aggregate function (either directly in its definition or indirectly by referring to a
mapped field).

Each field name may be qualified with either a file name or number that indicates which element
in the list of files, members, and record formats specified for the FILE parameter contains the
field. The specified value *MAPFLD may be used to qualify the field name if the field is defined
on the Mapped field specifications (MAPFLD) parameter.

For more information on data type compatibility, see Database information in the iSeries

---

**Key field specifications (KEYFLD)**

Specifies the name of one or more key fields that are used to arrange the query records, or specifies that
the access path sequence of the first or only file, member, and record format specified for the File
specifications (FILE) parameter is used to arrange the query records. If key field names are specified, you
also indicate whether the part of the key associated with each key field is ascending or descending, and
whether the records are arranged by the absolute value of a numeric key field. If the key field specified is
double-byte (DBCS) field, the data is arranged in a single-byte sequence.

**Single values**

*NONE*  
No key fields are used to arrange the query records; therefore, any arrangement is acceptable. It
is even possible for the system to give query records in a different arrangement if the same query
is run twice, based on such factors as the current number of records in the file members queried.
*NONE allows the system more flexibility to improve the performance of processing records
through the open query file.

*FILE*  
The query records have the same arrangement as the first file, member, and record format
specified for the File specifications (FILE) parameter. *FILE can be specified even if the first file
in the list has only an arrival sequence access path, in which case the query record arrangement
matches the arrival sequence of the first file, member, and record format specified for the FILE
parameter.

When KEYFLD(*FILE) is specified, and a sort sequence other than *HEX has been specified for
the SRTSEQ parameter, you may receive your records in an order that does not reflect the true
file order. If the file is keyed, the sort sequence is applied to the key fields of the file. If the file
has a sort sequence table or an alternative collating sequence table, ordering is ignored. This
allows users to indicate which fields to apply a sort sequence to without having to list all the
field names. If a sort sequence is not specified for the query, the query is ordered as in releases
previous to V2R3M0.

**Element 1: Key field**

Specify one or more field names (a maximum of 50 field names can be specified) to be used to define a
keyed access path to arrange the query records. Each field name may be qualified with either a file name
or number that indicates which element value in the list of files, members, and record formats specified
for the File specifications (FILE) parameter contains the field. The special value *MAPFLD may also be
used to qualify the field name if the field is defined on the Mapped field specifications (MAPFLD)
parameter.

The sum of the lengths of all key fields cannot be more than 10000 bytes. In addition, if the sum of the
lengths of the key fields is greater than 2000 bytes, *INP must be specified for theOPTION parameter and
fields cannot be ordered by their absolute value.
Note: The limits noted above are reduced by 2 bytes for each variable-length key field used. For instance, if three key fields are variable-length, the sum of the lengths of all key fields cannot exceed 9994 bytes, since 10000 bytes - (3 variable-length fields * 2 bytes per field) = 9994 bytes.

Qualifier 1: Key field

*name Specify the name of the field to be used as a key field.

Qualifier 2: File or element

*MAPFLD
The field is defined on the MAPFLD parameter.

1-32 Specify the position of the element list value for the FILE parameter to be used. The element list value identifies the database file, file member, and record format to be used.

*name Specify the name of a database file specified for the FILE parameter.

Element 2: Key field order

*ASCEND
The part of the key defined by the specified key field is ordered by ascending key values.

*DESCEND
The part of the key defined by the specified key field is ordered by descending key values.

Element 3: Order by absolute value

*ABSVAL
The part of the key defined by the specified key field is arranged by the absolute value of the key field. *ABSVAL is specified together with either *ASCEND or *DESCEND, but it is ignored if the key field is not numeric. If *ABSVAL is not specified, the records are arranged by the signed value of a numeric key field.

Unique key fields (UNIQUEKEY)

Specifies whether the query is restricted to records with unique key values, and specifies how many of the key fields must be unique. If *ALL or a number is specified for this parameter, null values are considered equal.

*NONE
The key fields specified for the Key field specifications (KEYFLD) parameter are not required to be unique. All query records are available through the open query file, regardless of key value.

*ALL  All key fields specified for the KEYFLD parameter must be unique. If there are multiple query records with the same values for all of the key fields, only the first such record is available through the open query file.

1-120 Specify the number of key fields, ranging from 1 through 120, that is unique. This value must be no larger than the number of key fields determined by the KEYFLD parameter. If there are multiple query records with the same value for the specified number of consecutive key fields, only the first such record is available through the open query file.
Join field specifications (JFLD)

Specifies whether the query joins records from multiple file members, and specifies how to join field values from the files, members, and record formats specified for the File specifications (FILE) parameter in constructing the query records.

The first file, member, and record format specified for the FILE parameter is called the join primary, and all other element list values specified for the FILE parameter are called join secondaries. This parameter specifies a list of pairs of field names, in which the first field in each pair provides a value that is used to select records in a join secondary that have the same value in the second field name of the pair.

The join from-field and to-field may be mapped fields (specified for the Mapped field specifications (MAPFLD) parameter), but you cannot use a field that depends on an aggregate function either directly in its definition or indirectly by referring to a mapped field.

The join from-field and to-field are not required to have identical field attributes. For more information on data type compatibility, see Database information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

If more than one file is specified for the FILE parameter, *NO is specified for the Join with default values (JDFTVAL) parameter and *ANY is specified for the Join file order (JORDER) parameter, then the system takes information from the Join field specifications (JFLD) parameter and the Query selection expression (QRYSLT) parameter and derives the final join specifications. If you specify a file on the FILE parameter that is not referred to on the QRYSLT parameter or the JFLD parameter, all records for that file are logically joined to all other records created from the other files specified for the FILE parameter.

If either *YES or *ONLYDFT is specified for the JDFTVAL parameter, or *FILE is specified for the JORDER parameter, the join fields must be specified for the JFLD parameter.

Up to 50 join field pairs can be specified.

Single values

*NONE

No join operation is specified. If more than one file is specified for the FILE parameter, *NO is specified for the JDFTVAL parameter, and *ANY is specified for the JORDER parameter, the system automatically finds the join fields from the QRYSLT parameter.

Element 1: From field

Specify a field name to provide the value used to select records in a join secondary file, member, and record format. The field name may be qualified with either a file name or number that indicates which element in the list of files, members, and record formats, specified for the FILE parameter contains the field. The special value *MAPFLD can also be used to qualify the field name if the field is defined on the MAPFLD parameter.

A join from-field is a simple field or a mapped field, defined on the MAPFLD parameter. If either *YES or *ONLYDFT is specified for the JDFTVAL parameter, a join from-field depends only on fields that are contained in the join primary or in join secondaries specified for the FILE parameter ahead of the join secondary associated with the to-field of the pair.

Qualifier 1: From field
name Specify the name of the from-field.

Qualifier 2: File or element

*MAPFLD
The field is defined on the MAPFLD parameter.

1-32 Specify the position of the element list value for the FILE parameter to be used. The element list value identifies the database file, file member, and record format to be used.

name Specify the name of a database file specified for the FILE parameter.

Element 2: To field

Specify a field name used to select records from a join secondary file, member, and record format in constructing the query records. The field name is qualified with either a file name or number that indicates which element in the list of files, members, and record formats specified in the FILE parameter contains the field. The special value *MAPFLD can also be used to qualify the field name if the field is defined on the MAPFLD parameter.

A join to-field is a simple field or a mapped field, defined on the MAPFLD parameter. If either *YES or *ONLYDFT is specified for the JDFTVAL parameter, a join to-field depends only on fields that are contained all in a single join secondary. If the join secondary is a join logical file, only fields contained in the primary physical file member for the join logical file are used as components of the join to-field. The sum of the lengths of all to-fields for each join secondary (after change, if the from-field and to-field attributes are not identical) cannot be more than 2000 bytes unless JDFTVAL(*NO) is specified, where there is no 2000-byte limit.

Qualifier 1: To field

name Specify the name of the to-field.

Qualifier 2: File or element

*MAPFLD
The field is defined on the MAPFLD parameter.

1-32 Specify the position of the element list value for the FILE parameter to be used. The element list value identifies the database file, file member, and record format to be used.

name Specify the name of a database file specified for the FILE parameter.

Element 3: Join operator

Specifies the type of join operation that is performed for the specified from-field and to-field. If *NO is specified for the JDFTVAL parameter and *ANY is specified for the JORDER parameter, or if more than one join field pair is specified, a different join operator may be specified for each pair. If *YES or *ONLYDFT is specified for the JDFTVAL parameter, or *FILE is specified for the JORDER parameter, then only one join operator may be specified regardless of the join pairs.

*EQ An equal join operation is performed.
*GT   A greater than join operation is performed.
*LT   A less than join operation is performed.
*NE   A not equal join operation is performed.
*GE   A greater than or equal join operation is performed.
*LE   A less than or equal join operation is performed.

---

### Join with default values (JDFTVAL)

Specifies whether the query file should include join records that use default values for any of the fields from a join secondary file that does not contain a record with correct field values that satisfy the join connections specified on the **Join field specifications (JFDL)** parameter.

Join processing attempts to collect field values from the join primary and join secondaries. It does so by matching join from-field values to records in a join secondary that produce the appropriate values in the join to-field. If there are no records in a join secondary to produce the to-field values required for the pairs of join fields associated with the join secondary, this parameter specifies whether query records should be constructed using default values for all fields obtained from the join secondary.

If the **File specifications (FILE)** parameter includes any join logic files, all join logical files must be compatible with this parameter’s value. If the data description specification (DDS) used to create a queried join logical file does not contain the JDFTVAL keyword, this parameter may not be used for any of the join logical files specified for the FILE parameter, and JDFTVAL(*NO) is required. If any join logical file has the JDFTVAL keyword specified for the FILE parameter, then join logical files for this open query file must be created using the JDFTVAL keyword, and *YES is required. If any files on the FILE parameter are view logical files, then *NO must be specified this parameter.

If the JDFTVAL attribute is not compatible with the attributes of the join logical files processed, you can replace the join or view logical files specified for the FILE parameter with their based-on physical file members. You can provide the correct, additional from-field and to-field pairs on the JFDL parameter in order to join records from the physical file members in any way.

If more than one file is specified for the FILE parameter, and either *YES or *ONLYDFT is specified, the system uses the join fields as specified for the JFDL parameter as the final join specification.

- **NO**   No default values are used to construct join query records.
- **YES**  Create all records for the join, including those produced both with and without using default values. No view logical files are allowed on the FILE parameter.
- **ONLYDFT**  Create only the records produced by using default values in constructing the join. This option is used to include only exception records in the records available through the open query file. If *ONLYDFT is specified, no join or view logical files may be specified for the FILE parameter.
Join file order (JORDER)

Specifies, for a join query, whether the join order must match the order specified for the File specifications (FILE) parameter. If the join order is varied, the query records are generated in a different arrangement. If the value specified for the Join with default values (JDTVAL) parameter is *YES or *ONLYDFT, this parameter is ignored. The order specified for the FILE parameter is always preserved, because changing the join order can change which records are returned when join default value processing is required.

If more than one file is specified in the FILE parameter and *FILE is specified, the system uses the join fields as specified for the Join field specifications (JFLD) parameter as the final join specifications.

*ANY Any join file order is allowed, and any such arrangement may be used by the system to create the query records. It is possible for a query to return result records in a different arrangement if the same query is run twice consecutively (based on factors such as the current number of records in the files that are asked). *ANY allows the system more flexibility to improve the performance of processing records through the open query file than any other Join file order (JORDER) parameter value.

*FILE The order of the file, member, and record format elements specified for the FILE parameter are preserved in the join operation.

Grouping field names (GRPFLD)

Specifies the field names that are used to group query results. One query record is created for each group of records (after join processing, if required) selected by the Query selection expression (QRYSLT) parameter. The group is defined by the collection of records that has the same set of values for the fields specified in the record format identified on the Format specifications (FORMAT) parameter. All null values within a group are considered equal. If no field names are specified and group processing is required, the whole file is considered to be one group. Each query record that is created is either made available through the open query file or is discarded, depending on the selection values specified for the Group selection expression (GRPSLT) parameter. To ensure a sequence, you must specify the Key field specifications (KEYFLD) parameter.

Single values

*NONE No fields are used to form groups. If the grouping function is required (because selection values are specified for the GRPSLT parameter, or an aggregate function is used by a field specified for the Mapped field specifications (MAPFLD) parameter), all records selected by the values specified for the QRYSLT parameter are handled as a single group.

Other values

Specify one or more field names (up to 50) to be used to group the query results. Each field name may be qualified with either a file name or number to indicate which element in the list of files, members, and record formats specified for the FILE parameter contains the field. The special value *MAPFLD may also be used to qualify the field name if the field is specified for the MAPFLD parameter.

A grouping field defined on the MAPFLD parameter cannot refer to an aggregate function in its definition (either directly, or indirectly through the use of another field specified for the MAPFLD parameter). The sum of the lengths of all grouping fields cannot exceed 2000 bytes.

Qualifier 1: Grouping field names
**name** Specify the name of a field to be used to group query results.

**Qualifier 2: File or element**

*MAPFLD  
The field is defined on the MAPFLD parameter.

1-32 Specify the position of the element list value for the FILE parameter to be used. The element list value identifies the database file, file member, and record format to be used.

**name** Specify the name of a database file specified for the FILE parameter.

---

**Group selection expression (GRPSLT)**

Specifies the selection values used after grouping to determine which records are available through the open query file.

*ALL All records defined by the grouping function described by the Grouping field names (GRPFLD) parameter are selected.

'group-selection'  
Specify an expression (contained in apostrophes) that describes the values used to determine which records are to be selected. Any logical expression formed from relationships (such as *EQ and *NE) of field and constant values, or functions of field and constant values, are specified. Only grouping fields (specified for the GRPFLD parameter), literals, aggregate functions (such as %AVG and %STDDEV), and mapped fields (specified for the Mapped field specifications (MAPFLD) parameter) that are composed of grouping fields, aggregate functions, and literals are referred to in any relationship. At least one field must be specified in each relationship.

Each field name may be qualified with either a file name or number that indicates which element in the list of files, members, and record formats specified for the File specifications (FILE) parameter contains the field. The special value *MAPFLD may also be used to qualify the field name if the field is specified for the MAPFLD parameter.

For more information on data type compatibility, see Database information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

---

**Mapped field specifications (MAPFLD)**

Specifies the definition of query fields that are mapped or derived from other fields. MAPFLD is generally not needed if the field names specified on other parameters are simple field names that exist in only one of the file, member, and record format elements specified for the File specifications (FILE) parameter.

Up to 50 mapped field definitions can be specified.

**Single values**

*NONE  
No mapped fields are needed. All field names specified on other parameters exist in some record format specified for the FILE parameter.

**Element 1: Mapped field**
**name**  Specify the simple field name used on any other parameter that must refer to this mapped field. A qualified name is **not** allowed for the first part of the parameter list element. All specified mapped-field-name values must be unique.

**Element 2: Field definition expression**

**character-value**

Specify an expression of up to 256 characters (contained in apostrophes) which defines the mapped field in terms of other fields that either exist in one of the file, member, and record format elements specified for the FILE parameter, or are defined by some other mapped field definition appearing earlier in the list. Either numeric operations or string operations are allowed, depending on the data type of the fields used in the definition.

Each field name may be qualified with either a file name or number that indicates which element in the list of files, members, and record formats specified for the FILE parameter contains the field. The special value *MAPFLD may also be used to qualify the field name if the field is specified for the **Mapped field specifications** (MAPFLD) parameter.

**Element 3: Mapped field type**

Specify the field type for this mapped field, or specify *CALC to allow the system to calculate appropriate attributes (including field type) for the mapped field. *CALC is the default if no field-type value is specified.

When *CALC is used, the field attributes are determined in one of two ways. The attributes either match the field definition in the record format identified on the **Format specifications** (FORMAT) parameter, or (if the field is not in the record format on the FORMAT parameter) the attributes are calculated based on the expression specified in the mapped-field-definition for this field. If the mapped field is used in the record format identified on the FORMAT parameter, you must either use *CALC or specify attributes (field-type, field-length, and field-decimals) identical to those of the field in the record format specified for the FORMAT parameter.

The field type must be valid for the final result of the expression specified for the mapped-field-definition.

The following are the mappings that are not supported between character, DBCS-open, DBCS-either, DBCS-only, graphic, binary string, and numeric types:

- From character or numeric to DBCS-only
- From DBCS-open to DBCS-either or DBCS-only
- From DBCS-either to character, numeric, or DBCS-only
- From DBCS-only or DBCS-graphic to character or numeric
- From UCS-2 or UTF-16 to DBCS-either or DBCS-only
- From binary string to any non-binary string
- From numeric to binary string

**Note:** Binary string refers to both BLOB and BINCHAR data types.

For more information on mappings see Database information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

*CALC  Calculate appropriate field type attributes.

*BIN2  Two-byte binary field.

*BIN4  Four-byte binary field.
*FLT4  Four-byte floating-point field.
*FLT8  Eight-byte floating-point field.
*DEC   Packed decimal field.
*ZONED Zoned decimal field.
*CHAR  Character field.
*VCHAR Variable length character field.
*HEX   Hexadecimal field.
*VHEX  Variable length hexadecimal field.
*DATE  Date field.
*TIME  Time field.
*TIMESTP  Timestamp field.
*ONLY  DBCS-only field.
*VONLY Variable length DBCS-only field.
*OPEN  DBCS-open field.
*VOPEN Variable length DBCS-open field.
*EITHER DBCS-either field.
*VEITHER Variable length DBCS-either field.
*GRAPHIC DBCS-graphic field.
*VGRAPHIC Variable length DBCS-graphic field.

Element 4: Length

0-32766

Specify the field length in number of digits for a numeric field, number of bytes for a character or DBCS field, or number of characters for a graphic field. A field length must be an even value for DBCS-only and DBCS-either field types. The range of valid lengths for each field type is shown in Table 1. A value must not be specified if *CALC is used for the element 3 (Mapped field type).
Table 1. Query Field Structure

<table>
<thead>
<tr>
<th>Field Type</th>
<th>External Field Length</th>
<th>Default Length and Decimals</th>
</tr>
</thead>
<tbody>
<tr>
<td>*BINZ</td>
<td>1-5</td>
<td>5 0</td>
</tr>
<tr>
<td>*BIN4</td>
<td>1-10</td>
<td>10 0</td>
</tr>
<tr>
<td>*FLT4</td>
<td>1-9</td>
<td>7 6</td>
</tr>
<tr>
<td>*FLT8</td>
<td>1-17</td>
<td>14 14</td>
</tr>
<tr>
<td>*DEC</td>
<td>1-63</td>
<td>15 5</td>
</tr>
<tr>
<td>*ZONED</td>
<td>1-63</td>
<td>15 5</td>
</tr>
<tr>
<td>*CHAR</td>
<td>1-32766</td>
<td>32</td>
</tr>
<tr>
<td>*VCHAR</td>
<td>0-32740</td>
<td>32</td>
</tr>
<tr>
<td>*HEX</td>
<td>1-32766</td>
<td>32</td>
</tr>
<tr>
<td>*VHEX</td>
<td>0-32740</td>
<td>32</td>
</tr>
<tr>
<td>*DATE</td>
<td>5-10</td>
<td>8</td>
</tr>
<tr>
<td>*TIME</td>
<td>4-8</td>
<td>7</td>
</tr>
<tr>
<td>*TIMESTP</td>
<td>14; 16-26</td>
<td>26</td>
</tr>
<tr>
<td>*ONLY</td>
<td>4-32766</td>
<td>32</td>
</tr>
<tr>
<td>*VONLY</td>
<td>0-32740</td>
<td>32</td>
</tr>
<tr>
<td>*OPEN</td>
<td>4-32766</td>
<td>32</td>
</tr>
<tr>
<td>*VOPEN</td>
<td>0-32740</td>
<td>32</td>
</tr>
<tr>
<td>*EITHER</td>
<td>4-32766</td>
<td>32</td>
</tr>
<tr>
<td>*VEITHER</td>
<td>0-32740</td>
<td>32</td>
</tr>
<tr>
<td>*GRAPHIC</td>
<td>1-16383</td>
<td>32</td>
</tr>
<tr>
<td>*VGRAPHIC</td>
<td>0-16370</td>
<td>32</td>
</tr>
</tbody>
</table>

Element 5: Decimal positions

0-63 Specify the number of decimal positions for a numeric field, expressed as a number of decimal digits, that is no larger than the total number of digits specified for the field length. If no value is given, the value is assumed to be zero. A value must not be specified for a binary or character field, or if *CALC is specified for element 3 (Mapped field type).

Element 6: Mapped field CCSID

*CALC The coded character set identifier (CCSID) value is determined by the CCSIDs of the fields or literal values that make up the MAPFLD field definition.

*HEX A pre-defined value is used such that no translation of the field data takes place.

1-65535 Specify the CCSID to be used. To see a complete list of identifiers when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt).

Literal values in the MAPFLD definition are tagged with the job default CCSID. However, if the MAPFLD consists of only a literal value and the user specifies a field-CCSID value, the literal value will be tagged with that CCSID. This allows you to tag a literal with a CCSID other than the job’s default CCSID.

Note: Normally, *HEX and *VHEX fields do not have an associated CCSID. Because of this, the data in the field is treated the same regardless of the default CCSID of the system on which the data is being used. However, if you specify a CCSID for a *HEX or *VHEX field, the CCSID overrides the hexadecimal attribute of the field (causing the field to be treated as *CHAR or *VCHAR), and the data in the field may be treated differently if it is moved to a system that has a different default CCSID.

Ignore decimal data errors (IGNDECERR)

Specifies whether the system ignores decimal data errors during query processing.
*NO The system does not ignore decimal data errors.

*YES The system ignores decimal data errors. When errors in decimal data are encountered, the not valid sign or digits are automatically changed to valid values.

Open file identifier (OPNID)

Specifies the identifier used to name the open query file so that it is referred to on the Close File (CLOF) or Position Database File (POSDBF) command when it is closed. The identifier must differ from the identifier associated with any other file previously opened with the Open Database File (OPNDBF) command or OPNQRYF command, and which is not yet closed.

*FILE The name of the first or only file specified for the File specifications (FILE) parameter is used for the open identifier.

ame Specify the name you want to associate with this open query file.

Limit to sequential only (SEQONLY)

Specifies whether sequential-only processing is used for the file, and specifies the number of records processed as a group when read or write operations are performed to the open query file. The open query file ODP uses a different SEQONLY value than the one specified on this parameter, depending on other parameter values specified on this command. A message is sent if the SEQONLY value is changed.

Single values

*NO The file does not use sequential-only processing.

Element 1: Sequential only

*YES The open query file uses sequential-only processing.

Element 2: Number of records

1-32767 Specify the number of records that are processed as a group when read or write operations are performed to the open query file. If no value is specified, the system calculates the number of records to be processed as a group.

Commitment control active (COMMIT)

Specifies whether this file is placed under commitment control.

Before a database file is opened under commitment control, the user must ensure that all files in the commitment transaction are journaled. If only the after images are being journaled, the system implicitly begins journaling both the before and the after images for the duration of the changes being made to files opened under this commitment definition.

*NO The open query file is not placed under commitment control.

*YES The open query file is placed under commitment control.
Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

**Note:** This parameter is not valid when TYPE is also specified.

*ACTGRPDFN

The scope of the open operation is determined by the activation group of the program that called the OPNQRYF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller. In a multithreaded job, only those opens within the same thread and within the same activation group can share this ODP.

*ACTGRP

The scope of the open data path (ODP) is the activation group. Only those shared opens from the same activation group can share this ODP. This ODP is not reclaimed until the activation group is deactivated, or until the Close File (CLOF) command closes the activation group.

*JOB

The scope of the open operation is the job in which the open operation occurs. If the job is multi-threaded, only those opens from the same thread can share this ODP.

Duplicate key check (DUPKEYCHK)

Specifies whether duplicate key checking should be done on input and output operations for the file opened by this command.

*NO  No duplicate key feedback is provided on input and output commands.

*YES Duplicate key feedback is provided on input and output commands.

Allow copy of data (ALWCPYDTA)

Specifies whether the system is allowed to copy data from the files, members, and record formats specified for the File specifications (FILE) parameter. If so, the system is allowed to open the query file to the copy. The system generally tries to avoid using a copy of the data because the copy does not reflect changes made to the database after the information is copied. However, certain requests require that the data be copied in order to perform the specified query functions (such as when key fields contained in multiple based-on files for a join are specified).

*YES  The system may use a copy of data from the files, members, and record formats specified for the File specifications (FILE) parameter. A copy of the data is used only when it is needed to perform the requested query functions.

*OPTIMIZE

The system uses a sort routine to order the output from the files, file members, and record formats specified for the FILE parameter. A sort routine is used only if the KEYFLD parameter is specified, and if using a sort routine would improve query performance without conflicting with other OPNQRYF options.

A sort will improve the performance of a query that returns most or all of the records in the file or files specified for the FILE parameter.
Using a sort can increase the time required for the OPNQRYF command to process. This occurs because the sort is performed and all records to be returned through the query are processed while the OPNQRYF command is active. However, because the records are already processed, the reading of the records (by using either a program or the CPYFRMQRYF command) is very fast. Therefore, the overall time to process the query is reduced.

Specifying the KEYFLD parameter for the OPNQRYF command does not ensure that the query will use an index if ALWCPYDTA(*OPTIMIZE) is specified. If a sort routine is used, the file is not opened with indexed access. If the program reading the records from the OPNQRYF command requires indexed access (random processing rather than sequential processing), ALWCPYDTA(*YES) or ALWCPYDTA(*NO) should be specified.

When a sort is used, the query file’s position is not changed when a ROLLBACK statement is issued. Therefore, when a query is opened that has parameters, ROLLBACK statements that follow do not reset the queried file’s position to where it was at the start of the unit of recovery.

Note: Do not specify ALWCPYDTA(*OPTIMIZE) if you require that a ROLLBACK statement reposition the query file, or if you require that the queried file be opened with indexed access.

The following items are required before a sort is valid for the OPNQRYF command:

- ALWCPYDTA(*OPTIMIZE) must be specified.
- The OPTION parameter, if specified, must be *INP.
- A value other than *FILE or *NONE must be specified on the KEYFLD parameter.
- The UNIQUEKEY parameter must not be specified, or must specify *NONE.
- The SEQUENCE parameter, if specified, must be *YES.
- The DUPKEYCHK parameter must not be specified, or must specify *NO.
- The total buffer length of all fields in the file specified for the FORMAT parameter (or FILE parameter, if the FORMAT parameter is not specified) must not exceed 32700 bytes.

The query optimizer determines whether a sort is used. This decision is based on the number of records expected from the query and the options specified for the OPNQRYF statement. The following items influence the optimizer’s choice of a sort:

- The OPTIMIZE parameter should specify *ALLIO or *MINWAIT. If *FIRSTIO is specified, the number of records specified should be close or equal to the number of result records expected from the query.
- The number of records in a file specified for the FILE parameter should contain a minimum of 200 records.
- The query result should contain a minimum of 200 records.

*NO The system does not use a copy of data from the files, members, and record formats specified for the File specifications (FILE) parameter. If it is necessary to use a copy of the data to perform the requested query functions, the query file is not opened and an error message is issued.

Performance optimization (OPTIMIZE)

Specifies what optimization goal is used by the system in deciding how to perform the selection and join processing necessary to satisfy other specifications on this command.

If the Key field specifications (KEYFLD) parameter or Grouping field names (GRPFLD) parameter require that an access path be built (when no existing access path can be shared), the access path is built completely, regardless of the value specified for this parameter. Optimization primarily affects the timing of selection processing.

Single values
**ALLIO**
The system attempts to improve the total time to process the whole query, assuming that all query records are read from the file.

**MINWAIT**
The system attempts to improve the query to minimize delays when reading records from the file.

Element 1: Performance optimization

**FIRSTIO**
The system attempts to improve the time to open the query file and retrieve the first buffer of records from the file.

Element 2: Number of records

1-2147483647
Specify the number of records expected to be retrieved. The query optimizer will use this information to determine the proper implementation for the query.

---

**Optimize all access paths (OPTALLAP)**

Specifies whether the query optimizer should consider all the access paths that exist over the files being queried when determining how to accomplish the query.

**NO**
Allow the query optimizer to operate normally. When determining how to start a query, the optimizer considers access paths until an internal timeout value has been exceeded. If there are a large number of access paths over the files being queried, the optimizer may time out before it has considered all the available access paths.

**YES**
Force the query optimizer to ignore the internal timeout value and consider all the available access paths over all the files in the query. Note that if there are a large number of access paths over the files it may take a long time to optimize the query.

---

**Sort sequence (SRTSEQ)**

Specifies the sort sequence to be used for sorting and grouping selections specified for the QRYSLT or GRPSLT parameters, joins specified for the JFLD parameter, ordering specified for the KEYFLD parameter, grouping specified for the GRPFLD parameter, %MIN or %MAX built in functions, or unique key values specified for the UNIQUEKEY parameter.

Single values

**JOB**
The SRTSEQ value for the job is retrieved for the job.

**HEX**
A sort sequence table is not used, and the hexadecimal values of the characters are used to determine the sort sequence.

**LANGIDSHR**
A shared weight sort table is used.

**LANGIDUNQ**
A unique weight sort table is used.

Qualifier 1: Sort sequence
name Specify the name of the sort sequence table to be used with this query.

Qualifier 2: Library

*LIBL All libraries in the user and system portions of the job’s library list are searched.

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

name Specify the name of the library to be searched.

Language ID (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

*JOB The LANGID value for the job is retrieved for the job.

language-ID Specify the language identifier to be used by the job.

Final output CCSID (CCSID)

Specifies the coded character set identifier (CCSID) in which data from character, DBCS-open, DBCS-either and graphic fields will be returned. Data from UTF-8, UCS-2, or UTF-16 fields will not be converted.

*JOB Data is returned in the CCSID of the job issuing the OPNQRYF command.

*HEX No CCSID conversion is performed before the data is returned.

1-65535 Specify a CCSID value. Data will be converted to this CCSID before it is returned.

Type of open (TYPE)

Specifies the level at which the Reclaim Resources (RCLRSC) command closes the file.

Note: This parameter is ignored unless the default value is specified on the OPNSCOPE parameter and the request is from the default activation group.

*NORMAL The Reclaim Resources (RCLRSC) command closes the file if the program call that ran this command is ended without closing the file.

*PERM The file remains open until the Close File (CLOF) command closes it, or until the routing step or default activation group ends. The query file remains open even if the Reclaim Resources (RCLRSC) command is run.
Examples

Example 1: Selecting Specific Records

Note: Additional examples of selecting records using the OPNQRYF command can be found in the Database Programming topic in the Information Center.

OPNQRYF FILE(ordfile) OPTION(+ALL)
  QRYSLT('orddate=%range("840101" "841231") &
  ordamt>100')
  KEYFLD((ordamt =descend))

This command uses the QRYSLT parameter to select only records in the first member of file ORDFILE that have an order date in 1984 and an order amount greater than 100. Because the FORMAT parameter is omitted, the open query file has the same record format as file ORDFILE. The open query file allows all file operations (input, output, update, and delete). The KEYFLD specification is used to force the records to be arranged by descending value of order amount.

Example 2: Using the %XLATE Built-In Function

OPNQRYF FILE(telefile)
  QRYSLT('%xlate(usrename qsysstrntbl) *ct
  "GEORGE"')

This command uses the %XLATE built-in function to translate the field USRNAME to uppercase, and to instruct the *CT operator to select only records that contain the value GEORGE in the field USRNAME. QSYSSTRNTBL is an IBM-supplied system translation table that converts lowercase alphabatics (a through z) to uppercase (A through Z). The translation is done to ensure that the search value is recognized even if its characters appear in mixed case. The records available through the open query file have the same record format as those in file TELEFILE.

Example 3: Using the %XLATE Built-In Function

OPNQRYF FILE(telefile)
  QRYSLT('usrename *ct "GEORGE"')
  MAPFLD((usrename
  '%xlate(telefile/usrename qsysstrntbl)'))

In the previous example, the value of field USRNAME, which is returned to the high-level language (HLL) program that reads records from the open query file, is not translated to uppercase. This example shows a way to make the uppercase version of field USRNAME available to the HLL program. This is done by defining a mapped field (MAPFLD parameter) for the translated value of field USRNAME. The field has the same field name as the field name in the open query file record format being used. The translated version of the field is used for selection (QRYSLT parameter) and is used in the open query file record format.

Example 4: Using the %SST Built-In Function

OPNQRYF FILE((histlib/ordfile hist1))
  OPTION(+inp +upd +dlt)
  FORMAT(ordinfo orddtls) QRYSLT('month=7')
  MAPFLD((year '%'sst(orddate 1 2) 'zoned 2)
  (month '%'sst(orddate 3 2) 'zoned 2)
  (day '%'sst(orddate 5 2) 'zoned 2))

This command uses the %SST built-in function to create a substring of the year, month, and day parts of character field ORDDATE in file ORDFILE. If the file ORDINFO has a record format, ORDDTLS, containing at least the field’s YEAR, MONTH, and DAY records, these fields have input-only usage in the open query file record format because they are defined by using a built-in function (%SST) and are mappings that mix character and numeric (zoned decimal format) types. The file is opened for input, update, and delete operations, but none of the field’s YEAR, MONTH, and DAY records are updated.
using the open query file open data path (ODP). The open query file uses only records in the HIST1 member of file ORDFILE in library HISTLIB, and the records retrieved through the file have the same format as record format ORDDTLS in file ORDINFO. Only records pertaining to the month of July are processed through the open query file (QRYSLT parameter).

**Example 5: Returning the First Record of Each Set**

```
OPNQRYF FILE((routeif *first locusr))
  QRYSLT('%sst(101 4) *eq "ROCH"')
  KEYFLD(fromusr fromloc tousr toloc) UNIQUEKEY(*all)
```

This command uses the KEYFLD and UNIQUEKEY parameters to return only the first record of each set of records in record format LOCUSR in the first member of file ROUTELF that have the same values for the fields FROMUSR, FROMLOC, TOUSR, and TOLOC. The query result is further restricted by selecting only records that have the value ROCH in the first four characters of field TOLOC. The records available through the open query file contain all of the fields in record format LOCUSR of file ROUTELF. If the file ROUTELF contains information about messages routed by an application, this example identifies all unique sender and receiver pairs in which the receiving location name begins with ROCH.

**Example 6: Joining a File to Itself**

```
OPNQRYF FILE(partpf partpf) FORMAT(partjoin)
  JFLD((1/pnbr 2/pnbr *GE))
  MAPFLD((pnm1 '1/pname')
        (pnm2 '2/pname')
        (pnbr '1/pnbr'))
```

This example illustrates how a file is joined to itself, as well as how to use the MAPFLD parameter to rename fields in the based-on files. A greater than or join is performed using field PNBR as both the join from-field and the join to-field.

The format of file PARTJOIN is assumed to contain fields named PNBR, PNM1, and PNM2. The field name PNBR is valid in the query output record format because that field is defined on the MAPFLD parameter. If the record format in file PARTJOIN contains a field named PNAME, an error occurs because the field exists in both files specified on the FILE parameter, and is not the name of a field defined on the MAPFLD parameter. The mapped field definitions are field names, so the attributes of fields PNM1 and PNM2 match the attributes of field PNAME, and the attributes of field PNBR in the open query file records match field PNBR in file PARTPF. Further, when a file is joined to itself, it is always necessary to specify a file number name for any field that is defined in the based-on file.

**Example 7: Renaming Fields in Based-On Files**

The same query can also be specified as follows:

```
OPNQRYF FILE(partpf partpf) FORMAT(partjoin)
  QRYSLT('1/pnbr *GE 2/pnbr')
  MAPFLD((pnm1 '1/pname')
        (pnm2 '2/pname')
        (pnbr '1/pnbr'))
```

Because more than one file is specified on the FILE parameter, and the default value is specified for the JDFTVAL and JORDER parameters, the system takes the join specifications from the values specified on the QRYSLT parameter.

**Example 8: Selecting Master Records With No Detail Records**

```
OPNQRYF FILE(cusmas ordfil) FORMAT(cusmas)
  JFLD((cusnbr ordfil/cusnbr)) JDFTVAL(*onlydft)
  MAPFLD((cusnbr 'cusmas/cusnbr'))
```
This command uses a join query to select only master records that have no associated detail records. The master file (CUSMAS) is joined (equal join) to the detail file (ORDFIL) by the customer number field that appears in both record formats. The customer number field name is the same in both record formats (CUSNBR). Because CUSNBR is the name of a field defined on the MAPFLD parameter, everywhere the simple field name CUSNBR is used, the mapped field version of the CUSNBR field in file CUSMAS is used (including the open query file record format, which matches the customer master file record format). The JDFTVAL parameter indicates that only records that are produced by using default values are available through the open query file. Every master record that has associated detail records (with the same value of the customer number field) is excluded, and every master record that has no associated detail records creates a result record.

**Example 9: Identifying Detail Records With No Associated Master Record**

```opnqryf
OPNQRYF FILE(ordfil cusmas) FORMAT(ordfil)
  JFLD((cusnbr cusmas/cusnbr)) JDFTVAL(*onlydft)
  MAPFLD((cusnbr 'ordfil/cusnbr'))
```

This change of the previous example (using the same files) shows how to identify all detail records with no associated master record (in this case, all orders with an unregistered customer number):

**Example 10: Calculating Basic Statistics**

```opnqryf
OPNQRYF FILE(scores) FORMAT(clsstats) GRPFLD(clsid)
  GRPSLT('clsavg<70 & clsmax-clsmin>30')
  MAPFLD((clsid 'count')
    (clsavg 'avg(usrscore)')
    (clsmin 'min(usrscore)')
    (clsmax 'max(usrscore)'))
```

This command uses the grouping function to calculate basic statistics for each group of records in file SCORES that have the same value in the field CLSID. Assuming file CLSTSTATS has a record format containing field CLSID and all fields specified on the MAPFLD parameter, each record available through the open query file contains the value of the grouping field (CLSID) as well as the number of records included in the group and the average, minimum, and maximum values of field USRSCORE in the group. Selection occurs after grouping, so that records are created for groups only when the average value of USRSCORE in the group is less than 70 and the difference between the maximum and minimum scores in the group is greater than 30.

**Example 11: Selecting Records With a Specific Value**

```opnqryf
OPNQRYF FILE(ITMMAST)
  QRYSLT('itmcode=range(32 50) & itmtype="P"')
  ALWCPYDTA(*NO) OPTIMIZE(*FIRSTIO)
  SEQONLY(*YES 10) TYPE(*PERM)
```

This command selects from the first member of file ITMMAST only the records that have a value of field ITMCODE in the range from 32 through 50 and also have a value of field ITMTYPE equal to the letter P. The ALWCPYDTA parameter specifies that the open query file must never use a copy of the records in file ITMMAST. The OPTIMIZE and SEQONLY parameter values cause the system to attempt to improve processing for the open query file to minimize the time needed to retrieve the first buffer of ten records. This combination of parameter values is a good choice if the file is used with a high-level language interactive inquiry program that shares the open query file open data path (ODP) and shows ten records on each display screen. The open data path (ODP) for the open query file is 'permanent' (TYPE parameter), which means that it remains open either until the file is closed by using the Close File (CLOF) command or until the routing step ends.

**Example 12: Tagging a Literal with a Specific CCSID**

```opnqryf
OPNQRYF FILE(itmmast) QRYSLT('itmtype=pfield')
  MAPFLD((pfield 'P' *CHAR 1 *N 930))
```
This command selects from the first member of file ITMMAST only the records that have a value of field ITMTYPE equal to the letter ‘P’ in character set 930. The mapped field is created so that the literal ‘P’ can be tagged with a specific CCSID.

If a literal is not tagged with a specific CCSID, it is assigned the CCSID of the job running the query. Because of this, if an OPNQRYF statement is part of a CL program that is shared among systems with differing CCSIDs (in different countries, perhaps), a query that uses a literal in the selection specifications may not return the same results on all systems, even though the data in the files is the same. This happens because the internal representation of the literal may be different when the CL program is run in a job with a different CCSID. This representation then may not match the same records in the file. Note that the internal representation of the data in the file does not change based on the CCSID of the current job.

Tagging the literal with a specific CCSID avoids this problem. A literal tagged with a specific CCSID keeps the same internal representation on all systems. The CCSID that is used to tag the literal should be the same as the CCSID assigned to the field against which the literal is being compared.

**Example 13: Using a Nonjoin Query**

```
OPNQRYF FILE((EMPLOYEE)) KEYFLD((NAME))
    ALWCPYDTA(*OPTIMIZE)
```

This command returns all of the records in the EMPLOYEE file.

**Example 14: Using a Join Query**

```
OPNQRYF FILE((EMPLOYEE) (MANAGEMENT)) FORMAT(EMPLOYEE)
    KEYFLD((NAME)) JFLD((1/EMPID 2/MEMPID))
    ALWCPYDTA(*OPTIMIZE)
```

This command returns all of the records required by the join criteria.

**Example 15: Query Comparing Character and Numeric Data**

```
OPNQRYF FILE((STAFF)) QRYSLT('SALARY > "18357.50"')
```

This command returns all of the records in the STAFF file where their salary is greater than 18357.50 even though SALARY is a numeric field and the literal value in the QRYSLT is character.

---

**Error messages**

*ESCAPE Messages*

**CPF2115**

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

**CPF2169**

Job’s sort sequence information not available.

**CPF2619**

Table &1 not found.

**CPF3BCC**

Language identifier &l not valid.

**CPF3BC6**

Sort sequence &l not valid.
CPF3BC7
CCSID &1 outside of valid range.

CPF3BC8
Conversion from CCSID &1 to CCSID &2 is not supported.

CPF3BC9
Conversion from CCSID &1 to CCSID &2 is not defined.

CPF3BDD
Sort sequence &1 not valid for UCS2 data.

CPF3FC0
Language identifier is not valid.

CPF4174
OPNID(&4) for file &1 already exists.

CPF8133
Table &4 in &9 damaged.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9813
Record format &3 in file &1 not found.

CPF9815
Member &5 file &2 in library &3 not found.

CPF9820
Not authorized to use library &1.

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

CPF9826
Cannot allocate file &2.

CPF9830
Cannot assign library &1.

CPF9899
Error occurred during processing of command.

*STATUS Messages
CPI4011
Query running. &2 records selected, &1 processed.

CPI4301
Query running.

CPI4302
Query running. Building access path for &2 in &1.

CPI4303
Query running. Creating copy of file &1 in &2.

CPI4304
Query running. &1 records selected. Selection complete.

CPI4305
Query running. Sorting copy of file *N in *N.

CPI4306
Query running. Building access path from file &1 in &2.

CPI4307
Query running. Building hash table from &2 in &1.
Otherwise (OTHERWISE)

Where allowed to run:
- Batch program (*BPGM)
- Interactive program (*IPGM)

Threading: Yes

Specifies the command or group of commands (in an If or Do group) that are processed if none of the conditions on any of the When commands within a Select command group were evaluated to be true. After the command or Do group is processed, control is passed to the next command after the End Select command associated with this Otherwise command. If the command specified in this parameter is a DO, DOWHILE, DOUNTIL, or DOFOR command, all commands within the Do group are considered to be the command specified by the parameter.

Restrictions:
- This command is valid only within a CL procedure.
- This command is valid only within a SELECT-ENDSELECT command group.
- Only one OTHERWISE may be specified in a SELECT-ENDSELECT command group.
- All WHEN commands in a SELECT-ENDSELECT command group must appear before the OTHERWISE command.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMD</td>
<td>Command</td>
<td>Command string</td>
<td>Optional, Positional 1</td>
</tr>
</tbody>
</table>

Command (CMD)

Specifies the command or commands (in a If or Do group) to be processed if no When commands had an expression that evaluated to true.

If the command specified in this parameter is a DO, DOWHILE, DOUNTIL, or DOFOR command, all of the commands specified within the Do group are considered to be part of the command specified by the parameter.

If no command is specified on the CMD parameter (a null OTHERWISE) control is passed to the next command after the ENDSELECT command associated with this WHEN command.

Any CL command can be specified on the CMD parameter, except the following commands:
- ELSE
- PGM, ENDPGM
- ENDDO
- MONMSG

© Copyright IBM Corp. 1998, 2006
• DCL, DCLF
• WHEN, OTHERWISE, ENDSELECT

Examples

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

The OTHERWISE specifies the command to run if none of conditions on any of the WHEN commands in a SELECT command group command group were matched. In this example the CHGVAR will be run when the value of &NAME is not *CMD and not *PGM.

Error messages

None
Override with Data Base File (OVRDBF)

Where allowed to run: All environments (*ALL)

Threadsafe: Conditional

The Override with Database File (OVRDBF) command is used to (1) override (replace) the file named in the program, (2) override certain parameters of a file that are used by the program, or (3) override the file named in the program and override certain parameters of the file being processed. Parameters overridden by this command are specified in the file description, in the program, or in other previously issued file override commands. This command applies to physical files, logical files, and distributed data management (DDM) files.

To override (replace) a file named in the program, specify the name of that file in the FILE parameter, and specify the name of the file that overrides it (the file to be processed by the program) in the TOFILE parameter. The other parameters of this command can be used to override parameter values contained in the file description of the overriding file.

To override only certain parameters of the file named in the program, instead of replacing the entire file, specify the name of the file in the FILE parameter and specify the *FILE value for the TOFILE parameter. Then use the other parameters of this command to override specific parameters of the file. Parameters that are not specified do not affect parameters specified in the file description, in the program, or in other previously issued file override commands.

Restrictions:

1. In a multithreaded job, this command may only be issued from the initial thread.
2. In a multithreaded job, only Activation Group or Job scoped overrides will affect opens performed in a secondary thread.

Note: The override cannot be used for all commands. A list of the commands that cannot be overridden, along with more information on overriding files is in the Files and file systems topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>File being overridden</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>TOFILE</td>
<td>Overriding to data base file</td>
<td>Single values: *FILE, Other values: Qualified object name</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Overriding to data base file</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>MBR</td>
<td>Overriding member</td>
<td>Name, *FIRST, *LAST, *ALL</td>
<td>Optional, Positional 3</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>POSITION</td>
<td>Starting position in file</td>
<td>Single values: *NONE, *START, *END</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: *Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: *RRN-rcd nbr</td>
<td>*KEY-nbr key flds</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unsigned integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 3: *KEY-rec format having key</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 4: *KEY-key value</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td>RCDFMTLCK</td>
<td>Record format lock</td>
<td>Values (up to 32 repetitions): *Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Record format</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>FRCRATIO</td>
<td>Records to force a write</td>
<td>Integer, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTSLR</td>
<td>Rcd format selector program</td>
<td>*Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Rcd format selector program</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>*Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>WAITFILE</td>
<td>Maximum file wait time</td>
<td>*Integer, *IMMED, *CLS</td>
<td>Optional</td>
</tr>
<tr>
<td>WAITRCD</td>
<td>Maximum record wait time</td>
<td>*Integer, *IMMED, *NOMAX</td>
<td>Optional</td>
</tr>
<tr>
<td>NBRRCDS</td>
<td>Records retrieved at once</td>
<td>*Integer</td>
<td>Optional</td>
</tr>
<tr>
<td>EODLY</td>
<td>EOF retry delay in sec</td>
<td>1-99999, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>LVLCHK</td>
<td>Record format level check</td>
<td>*NO</td>
<td>Optional</td>
</tr>
<tr>
<td>EXPCHK</td>
<td>Check expiration date</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>INHWRT</td>
<td>Inhibit write</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>SECURE</td>
<td>Secure from other overrides</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OVRSCope</td>
<td>Override scope</td>
<td>*ACTGRPDFN, *CALLLVL, *JOB</td>
<td>Optional</td>
</tr>
<tr>
<td>SHARE</td>
<td>Share open data path</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OPNSCOPE</td>
<td>Open scope</td>
<td>*ACTGRPDFN, *JOB</td>
<td>Optional</td>
</tr>
<tr>
<td>SEQONLY</td>
<td>Limit to sequential only</td>
<td>Single values: *NO</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: *Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Sequential only</td>
<td>*YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Number of records</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>DSTDATA</td>
<td>Distributed Data</td>
<td>*BUFFERED, *PROTECTED, *CURRENT</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**File being overridden (FILE)**

Specifies the file in the using program to which this override command is applied. The specified file must be a database file when *FILE is specified in the Overriding to data base file (TOFILE) parameter. Otherwise, any device file or database file name can be specified.

This is a required parameter.

**name** Specify the name of the file.
Overriding to data base file (TOFILE)

Specifies the database file that is used instead of the file specified on the File being overridden (FILE) parameter, or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified on this command override the same parameters specified in the database file, in the program, or in other previously issued OVRDBF commands.

Single values

*FILE The database file named in the File being overridden (FILE) parameter has some of its parameters overridden by values specified in this command.

Qualifier 1: Overriding to data base file

name Specify the name of the database file that is used instead of the file specified in the FILE parameter.

Qualifier 2: Library

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

*CURLIB

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

name Specify the library where the database file is located.

Overriding member (MBR)

Specifies the members used within the database file. This parameter is not valid for distributed data management (DDM) files that refer to remote systems other than the System/38 or the AS/400 system.

*FIRST

The first member of a database file is used.

*LAST

The last member of a database file is used.

*ALL All members in your file are processed in order. All members are opened with the same override parameters as the first member. Overrides issued prior to the open of the first member are processed, but overrides or delete overrides issued following the open of the first member are not processed. EOFLDLY, FMTSILR, INHWRT, or the POSITION parameter cannot be specified if MBR(*ALL) has been specified on a previously issued OVRDBF command that is still in effect for this file. An escape message is sent if any of the mutually exclusive parameters are specified.

name Specify the member name that overrides (at file open time) the member name specified in the using program, or in other called OVRDBF commands. If the member name is not specified, and a TOFILE parameter other than *FILE has been specified, the first member in the file is used.
Starting position in file (POSITION)

Specifies the starting position for reading records from the database file. The first record to get can be at the beginning (*START) or at the end (*END) of the file, the nth record in the file (*RRN), or the record indicated by a key field value and one of the key-search values (*KEY, *KEYA, *KEYAE, *KEYB, or *KEYBE). This parameter overrides the value specified in the program, or in other called OVRDBF commands.

Note: This parameter cannot be specified if *ALL was specified previously on the Overriding member (MBR) parameter.

Single values

*NONE

No special positioning is required. The first input/output operation indicates the record that is read.

*START

The starting position is the first record in the file. If a read-previous record operation is specified in the program, an end-of-file condition occurs.

*END

The starting position is the last record in the file. When the next record is read, an end-of-file condition is reached. If a read previous record operation is requested, the last record of the file is read.

Element 1: Retrieve order

*RRN

The starting position is the relative record number specified for the second element of this parameter.

*KEYB

A record that precedes the record identified by the remaining search values (number of fields, record format name, and key value) is the first record read.

*KEYBE

The record identified by the search values is the first record read. If no record matches those values, the record that matches the largest previous value is selected.

*KEY

The record identified by the search values is the first record read. If a read-previous record operation is specified in the program, the preceding record is read.

*KEYAE

The record identified by the search values is the first record read. If there is no record that matches those values, the record with the next highest value is selected.

*KEYA

A record that follows the record identified by the remaining search values (number of fields, record format name, and key value) is the first record read.

Element 2: *RRN-rcd nbr *KEY-nbr key flds

relative-record-number

Specify the relative record number (its position from the beginning of the file) of the record that is read first. The value *RRN must be specified for the first element of this parameter. For example, POSITION(*RRN 480) specifies that record 480 is read next. If a read-previous record operation is requested, the 479th record in the file is read.

number-of-key-fields

Specify the number of key fields to use in the search, if *KEYB, *KEYBE, *KEY, *KEYAE, or *KEYA is specified for the first element of this parameter. The number of fields specified does not have to be the same as the actual number of fields in each key for the file. For example, if you
specify POSITION(*KEY 1 FMT1 A), the first record in the file format FMT1 that has a first key field value of A is read. If you specify a key value of zero, the search is based on all key fields. If zero is used, the key value contains the maximum key size. If it does not, no match occurs.

Element 3: *KEY-rec format having key

**name**  Specify the name of the record format in the database file that contains the key value specified. If no record format name is specified, all record formats are searched for the first record that matches the other search values.

Element 4: *KEY-key value

**character-value**  Specify the first record read. The key value is specified as a character string enclosed in apostrophes for character or positive zoned decimal formats, or is specified in hexadecimal form (\x'value\'). You can specify up to 2000 characters in the character string.

For example, POSITION(*KEY 1 FMT2 X'123F') specifies that:

1. The system searches for a record from the record format FMT2.
2. A single key field is used in the search (even though the key value may have more key fields).
3. The record contains the hexadecimal value 123F (the hexadecimal equivalent of packed decimal value 123.0). You get this record when it is found.

The Distributed Data Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter has more information on the effects of using the POSITION parameter with DDM files.

---

**Record format lock (RCDFMTLCK)**

Specifies the lock state of the named record format while it is used by the program. The lock state indicates how the data associated with each format is locked. The following example shows the lock states that are specified for each record format and the operations allowed to other programs when the lock is in effect:

<table>
<thead>
<tr>
<th>Lock State</th>
<th>Other Program Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>*SHRRD (Shared read)</td>
<td>Read and update allowed</td>
</tr>
<tr>
<td>*SHRNUP (Shared read, no update)</td>
<td>Read allowed, update not allowed</td>
</tr>
<tr>
<td>*SHRP (Shared update)</td>
<td>Read and update allowed</td>
</tr>
<tr>
<td>*EXCLRD (Exclusive allow read)</td>
<td>Read allowed, update not allowed</td>
</tr>
<tr>
<td>*EXCL (Exclusive no read)</td>
<td>Neither read nor update allowed</td>
</tr>
</tbody>
</table>

An explanation of each lock state is in the CL information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

For each record format, specify the record format name, followed by one lock state value. If the lock state specified for the file in an Allocate Object (ALCOBJ) command is more restrictive than the lock state specified in this parameter, this parameter is ignored. Thus, this parameter can only impose a more restrictive lock state on a record format than the lock state specified for the file.

You can specify 32 values for this parameter.

**Element 1: Record format**
name Specify the name of record format.

Element 2: Lock state

lock-state Specify one of the lock state values from the above table.

Records to force a write (FRCRATIO)

Specifies the number of insert, delete, or update operations that can occur on records before those records are forced into auxiliary (permanent) storage. If this physical file is being journaled, either a large number or *NONE should be used. *NONE may cause long synchronization of the journal and physical files. More information on this parameter is in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter, Appendix A. More information on journal management is in the Backup and Recovery book, SC41-5304.

This parameter overrides the force-write ratio specified in the database file, in the program, or in other previously issued OVRDBF commands.

*NONE
There is no force write ratio; the system determines when the records are written to auxiliary storage.

integer Specify the number of records written the changes are forced to disk. If a physical file associated with this database file is recorded in a journal, specify a larger force-write ratio.

Rcd format selector program (FMTSLR)

Specifies the record format selection program that is called when a logical file member contains more than one logical record format. The user-written selection program is called when a record is inserted into the database file and a record format name is not included in the high-level language program. More information about the use of format selector programs is in the Database information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter. This parameter overrides the value specified in the database file and in other previously issued OVRDBF commands.

A program specified as the format selector program cannot be created with USRPRF(*OWNER) specified in the Create CL Program (CRTCLPGM) command.

Note: This parameter cannot be specified if *ALL was specified previously on the Overriding member (MBR) parameter.

Qualifier 1: Rcd format selector program

name Specify the name of the selection program.

Qualifier 2: Library

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

*CURLIB The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.
name Specify the library where the program is located.

Maximum file wait time (WAITFILE)

Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

This parameter overrides the wait time specified in the database file, in the program, or in other previously issued OVRDBF commands.

More information on this parameter is in CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter, Appendix A.

*IMMED
   The program does not wait. When the file is opened, an immediate allocation of the file resources is attempted.

*CLS   The default wait time specified in the class description is used as the wait time for the allocation of the file resources.

integer Specify the number of seconds that the program waits for the allocation of the file resources. Valid values range from 1 through 32767 seconds.

Maximum record wait time (WAITRCD)

Specifies the number of seconds that a program waits for a record to be updated or deleted, or for a record read in the commitment control environment with LCKLVL(*ALL) specified. More information on record locking is in the Database information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter. If the record is not allocated in the specified wait time, an error message is sent to the program.

Note: This parameter overrides the record wait time specified in the database file, specified in the program, or in other previously issued OVRDBF commands. The minimum delay for DDM files is 60 seconds. This value may need to be longer than the delay specified for local database files.

*NOMAX
   The program waits indefinitely for a record lock.

*IMMED
   The program does not wait. An immediate lock of the record is obtained when the record is read.

integer Specify the number of seconds that the program waits for the record lock. Valid values range from 1 through 32767 seconds.
Records retrieved at once (NBRRCDS)

Specifies the number of records read from auxiliary storage as a unit and written to main storage as a unit. The amount of data actually read is equal to the number of records times the physical record length, not the logical record length.

This parameter is valid for sequential or random processing and is specified only when the data records are physically located in auxiliary storage in the sequence in which they are processed. This parameter overrides the number of records value specified in the program, or in other previously issued OVRDBF commands.

\textit{integer}

Specify the number of records. Valid values range from 1 through 32767.

EOF retry delay in sec (EOFDLY)

Specifies the number of seconds of delay before trying to read additional records when end of file is reached. This delay is used to allow other jobs an opportunity to add records to the file, and have the new records processed without having to start the job again. When the delay time ends, the job is made active, and data management determines whether any new records were added. If no new records were added, the job waits for another time delay without informing the application program. When a number of seconds is given, no end of file occurs on the given database file until an End Job (ENDJOB) command or forced end of data (FEOD) occurs.

\textbf{Note:} This parameter cannot be specified if *ALL was specified previously on the \textbf{Overriding member (MBR)} parameter.

There are several ways to end a job that is waiting for records due to an EOFDLY. They are:

- Write a record to the specified file which is recognized by the application program as a last record. The application program may then do a force end of data (FEOD) to start the end-of-file processing or close the file.
- End the job using the controlled value (ENDJOB OPTION(*CNTRLD)) with a delay time greater than the time specified on the EOFDLY time. The DELAY parameter time specified must allow for the EOFDLY time to run out, plus time to process any new records that may have been added to the file, and any end-of-file processing that is done in the user’s application. The end-of-file is set by database, and a normal end-of-file condition occurs after new records are retrieved.
- End the job immediately (ENDJOB OPTION(*IMMED)).
- If the job is interactive, start a system request and end the previous request.

*\textbf{NONE}*

Normal end-of-file processing is done.

1-99999

Specify the number of seconds that the program waits between attempts to get a record when an end of file condition occurs. No end of file is signaled until force end of data occurs, or until the job is ended with the *CNTRLD option.
Record format level check (LVLCHK)

Specifies whether the level identifiers for the record formats of the database file are checked when the file is opened by a program. For this check, which is done while the member is opened, the system compares the record format identifiers of each record format used by the program with the corresponding identifiers in the database member. Level checking cannot be done unless the program contains the record format identifiers. This command cannot override level checking from *NO to *YES.

*NO The level identifiers are not checked when the file is opened.

Check expiration date (EXPCHK)

Specifies whether the expiration date of the named member is checked. This date check is valid only on a physical file member. This parameter overrides the value specified in the program, or in other called OVRDBF commands.

*YES The expiration date of the physical file member is checked. If the current date is later than the expiration date, an escape message is sent to the program.

*NO The expiration date is not checked.

Inhibit write (INHWRT)

Specifies whether the processed records are written, deleted, or updated in the database file. The inhibit write parameter allows you to test a program without storing the processed records in the database. This parameter overrides the INHWRT parameter in other previously issued OVRDBF commands.

Note: This parameter cannot be specified if *ALL is specified on the Overriding member (MBR) parameter.

*YES Processed records are prevented from being written into the database. They are written only to an output device.

*NO All new and changed processed records are written into the database unless the program is in debug mode with *NO specified on the Update production files (UPDPROD) parameter, and the file is in a production library. In that case, an escape message is sent to the program.

Secure from other overrides (SECURE)

Specifies whether this file is safe from the effects of previously called file override commands.

*NO This file is not protected from other file overrides. Its values are overridden by the effects of any file override commands that were previously called.

*YES This file is protected from the effects of any file override commands that were previously called.

Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.
**ACTGRPDFN**

The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

**CALLLVL**

The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

**JOB**

The scope of the override is the job in which the override occurs.

---

**Share open data path (SHARE)**

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the Database information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

**NO**

The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

**YES**

If the member is opened more than once, the same ODP is shared with each program in the job that also specifies *YES on the Share open data path (SHARE) parameter when it opens the member. This includes several open operations in the same program.

---

**Open scope (OPNSCOPE)**

Specifies the extent of influence (scope) of the open operation.

**ACTGRPDFN**

The scope of the open operation is determined by the activation group of the program that called the OVRDBF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

**JOB**

The scope of the open operation is the job in which the open operation occurs.

---

**Limit to sequential only (SEQONLY)**

Specifies, for database files whose records are processed in sequential order only, whether sequential only processing is used on the file. This parameter also specifies the number of records transferred as a group to or from the database if sequential only processing is used. If a number is not specified, a default number is determined by the system. This parameter is used to improve the performance of programs that process database files in a sequential manner. This parameter overrides the value specified in the program or in other previously issued OVRDBF commands.

For files opened for input only in a program, the specified number of records is transferred as a group from the database to an internal data management buffer.
For files opened for output only in a program, a group of records is transferred to the database whenever the internal data management buffer receives the specified number of processed records from the program. For output files, sequential-only processing is valid for physical file members and for logical file members that are based on one physical file member only.

If SEQONLY(*YES) is specified, and any of the following conditions are true, the SEQONLY parameter is ignored and a message is issued.

- The program opened the member for output only and SEQONLY(*YES) is specified with the default number of records, and the member opened is either a logical member, a unique keyed physical member, or other access paths are built over the physical member.
- The program opened the member for other than input or output.
- The member opened by the program for output is based on many other members.
- The record length plus the feedback area sum exceeded 32,767 bytes.

**Note:** Unpredictable results occur when this parameter is used for alternate index files for DDM on a system other than an iSeries or AS/400 system.

**Single values**

*NO The database file is not restricted to sequential only processing.

**Element 1: Sequential only**

*YES The database file uses sequential only processing. A default value for the number of records transferred as a group is determined by the system based on how the file is used, the type of access path involved, and the file’s record length:

- The default is approximately the number of records that fit in an internal buffer of 4K for:
  - All database files opened for input only
  - Physical files opened for output that are only processed in either arrival sequence or in non-unique keyed sequence and that have no logical file members based on them
- The default is 1 record for:
  - All logical files opened for output only
  - Physical files opened for output only that either have unique keyed sequence access paths or have at least one dependent logical file with a keyed sequence access path that does not share the access path of the keyed physical file member

**Element 2: Number of records**

**integer**

Specify the number of records transferred each time. Valid values range from 1 through 32767. The file uses sequential only processing, and you must specify a value indicating the number of records in each group transferred between the database and the internal buffer. The user must ensure that the buffer size specified is always available to the program in the storage pool in which the program is running. The file uses sequential-only processing.

While records are in the internal data management buffer, other jobs can make changes to the same records in the database, and the program performing sequential-only input processing does not see the updates. To ensure that no other updating is done to records while they are in the buffer, the Allocate Object (ALCOBJ) command can be used in the program to specify either an *EXCLRDL or an *EXCL lock on the file.

If a program performs sequential-only output processing and does not handle output errors (such as duplicate keys and conversion mapping errors) that may occur when the records in the buffer are written to the database, records in the buffer after the first record in error are not written.
If the file is opened for output and the value specified in this parameter is not the same as the force write ratio specified for the file, the value used by the system is the smaller of the two; a message stating which value is changed is sent to the user.

When processing SEQONLY(*YES) for writing records into a database file, feedback information for each record (such as relative record number) is not always changed. If such feedback information is important, specify SEQONLY(*NO) or SEQONLY(*YES 1).

More information on sequence-only database files is in the Database information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

---

**Distributed Data (DSTDTA)**

Specifies the data retrieval method used for a distributed file. This parameter has no effect if used against a non-distributed file. Other parameters, such as SEQONLY, still affect how the data is retrieved from each system, and this parameter controls how all the data is managed when accessing a distributed file. This parameter overrides the distributed file data retrieval method selected by the system, or specified in other previously issued OVRDBF commands. More information on DSTDTA can be found in the DB/2 Multisystem for iSeries information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

*BUFFERED*

In order to achieve the best performance, data from the remote system and the local system may be kept in a buffer until retrieved by the user.

*PROTECTED*

Data can be buffered, but the file is locked to prevent updates by other jobs. This will give the same performance as *BUFFERED, but guarantees current data. While one job is using this option, other jobs will not be able to update the data in the file.

*CURRENT*

Data is not buffered. This option results in fully live data, with maximum concurrency, but without optimal performance.

---

**Examples**

**Example 1: Overriding An Existing Member**

OVRDBF FILE(ORDERSIN) MBR(MONDAY)

This command overrides the existing member with member MONDAY. With the override in effect, the member MONDAY will be processed when the file ORDERSIN is opened.

**Example 2: Overriding a Share Specification**

OVRDBF FILE(ORDERSIN) SHARE(*YES)

This command overrides the share specification for the file ORDERSIN. Because of this override, any subsequent opens of this file within the routing step share the ODP for the file.

**Example 3: Overriding a File, Member and Lock State**

OVRDBF FILE(INPUT) TOFILE(PAYROLL) MBR(MBR1)

RCDFMTLC({EMPODATA +EXCL})
This command overrides the file, the member, and the lock state of the record format EMPDATA. The override will cause the following to occur when the file INPUT is opened:

- The file PAYROLL will be processed instead of the file INPUT.
- The member MBR1 will be processed instead of the previously specified member.
- The lock *EXCL will be placed on record format EMPDATA instead of the existing lock. (*EXCL prevents another program from using the record format while the override is in effect.)

---

**Error messages**

*ESCAPE Messages*

CPF180C

   Function &1 not allowed.
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPHT (Merge TCP/IP Host Table)
Override with Display File (OVRDSPF)

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

The Override with Display File (OVRDSPF) command is used to: (1) override (replace) the file named in the program, (2) override certain parameters of a file that are used by the program, or (3) override the file named in the program and override certain parameters of the file processed. Parameters overridden by this command are specified in the file description, in the program, or in other called file override commands.

If a file named in the program is overridden, the name of that file is specified in the FILE parameter and the name of the overriding file (the file being processed) is specified in the TOFILE parameter. The OVRDSPF command also specifies parameters to override values contained in the file description of the overriding file. If the file named in the program is not replaced but certain parameters of the file are overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters overridden are then specified by the other parameters of the OVRDSPF command. Parameters that are not specified do not affect parameters specified in the file description, in the program, or in other called file override commands.

More information on override files is in the Application Display Programming book, SC41-5715.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>File being overridden</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
</tbody>
</table>
| TOFILE  | Overriding to display file | Single values: *FILE  
Other values: Qualified object name | Optional, Positional 2 |
|         | Qualifier 1: Overriding to display file | Name | |
|         | Qualifier 2: Library | Name, *LIBL, *CURLIB | |
| DEV     | Device | Values (up to 50 repetitions): Name, *REQUESTER | Optional, Positional 3 |
Other values: Element list | Optional |
|         | Element 1: Graphic character set | Integer | |
|         | Element 2: Code page | Integer | |
| DECFMT  | Decimal format | *FILE, *JOB | Optional |
| SFLENDTXT | SFLEND text | *MSG, *FILE | Optional |
| IGCDTA  | User specified DBCS data | *NO, *YES | Optional |
| IGCEXNCHR | DBCS extension characters | *YES, *NO | Optional |
| WAITFILE |Maximum file wait time | Integer, *IMMED, *CLS | Optional |

© Copyright IBM Corp. 1998, 2006
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAITRCD</td>
<td>Maximum record wait time</td>
<td>1-32767, *NOMAX, *IMMED</td>
<td>Optional</td>
</tr>
<tr>
<td>LVLCHK</td>
<td>Record format level check</td>
<td>*NO</td>
<td>Optional</td>
</tr>
<tr>
<td>SECURE</td>
<td>Secure from other overrides</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OVRSCOPE</td>
<td>Override scope</td>
<td>*ACTGRPDFN, *CALLLVL, *JOB</td>
<td>Optional</td>
</tr>
<tr>
<td>DTAQ</td>
<td>Data queue</td>
<td>Single values: *NONE&lt;br&gt;Other values: Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Data queue</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>SHARE</td>
<td>Share open data path</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>OPNSCOPE</td>
<td>Open scope</td>
<td>*ACTGRPDFN, *JOB</td>
<td>Optional</td>
</tr>
</tbody>
</table>

### File being overridden (FILE)

Specifies the file in the using program to which this override command is applied. The specified file must be a display device file when *FILE is specified in the Overriding to display file (TOFILE) parameter. Otherwise, any device file or database file name can be specified.

This is a required parameter.

**name**  Specify the name of the file.

### Overriding to display file (TOFILE)

Specifies the display file that is used instead of the file specified in the File being overridden (FILE) parameter, or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified on this command override the same parameters specified in the display device file, in the program, or in other called (OVRDSPF) commands.

#### Single values

**FILE**  The display device file named in the FILE parameter has some of its parameters overridden by values specified in this command.

**Qualifier 1: Overriding to display file**

**name**  Specify the name of the display device file that is used instead of the overridden 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 device file. If no current library entry exists in the library list, QGPL is used.

**name**  Specify the library where the device file is located.
Device (DEV)

Specifies the names of one or more display devices that are used with the display device file. This parameter overrides the device names specified in the device file, in the program, or in other called Override with Display File (OVRDSPF) commands. The device name specified in the IBM-supplied display device file is *REQUESTER.

This parameter overrides the device names specified in the device file, in the program, or in other called OVRDSPF commands.

*REQUESTER
  The display device from which the program is called is the device assigned to the file when the file is opened.

name
  Specify the names of one or more display devices that are used with this device file to pass data records between the users of the devices and the system. Each device name must already be known on the system (in a device description) before this device file is created. *REQUESTER can be specified as one of the names.

A maximum of 50 device names (including *REQUESTER, if it is specified) can be specified in this command, but the total number cannot exceed the number specified on the Maximum devices (MAXDEV) parameter when the file is opened.

Character identifier (CHRID)

Specifies the character identifier (graphic character set and code page) for the file. When a display file that was created with the CHRID DDS keyword is used with a work station device, the system translates data sent to and received from the device (as necessary) to ensure that the correct characters are displayed, and the correct hexadecimal byte values are returned to the application program. More information about display file CHRID processing and the translation tables that are used to convert data sent to and received from the display are in the Application Display Programming book, SC41-5715.

Single values

*DEVD
  The CHRID value specified in the device description of the work station on which the application is running is used. The *DEVD value means no translation is necessary because the file has the same character identifier as the work station. For a list of valid values, see the CHRID parameter of the Create Device Description Display (CRTDEVDSP) command.

*SYSVAL
  The CHRID value specified for the system on which the application is running is used. Translation may be necessary depending on the character identifier of the work station.

*JOBCCSID
  The character data is changed from the device CHRID to the CCSID (coded character set identifier) of the job on display file input, and from the CCSID of the job to the device CHRID on display file output. The character data is converted, if necessary, from the device CCSID (coded character set identifier) of the job during input, and from the CCSID of the job to the device CHRID on output.

Note: This value is not allowed if the file was created on a system at an earlier release level than V2R3M0.

*CHRIDCTL
  The system checks the CHRIDCTL job definition attribute to determine whether to use *JOBCCSID or *DEVD on the CHRID command parameter for this file.
Element 1: Graphic character set

integer

Specify the graphic character set value that matches the attributes of the display device. Valid values range from 1 through 32,767.

Element 2: Code page

integer

Specify the code page value that matches the attributes of the display device. Valid values range from 1 through 32,767.

Decimal format (DECFMT)

Specifies which decimal format value is used when editing numeric fields with the EDTCDE DDS keyword. The decimal format value determines the use of commas and periods for the decimal position and three digit positional separators on edited fields.

*FILE Use the decimal format value stored with the file when the file was created.

*JOB Use the decimal format value from the DECFMT job attribute when the file is opened.

SFLEND text (SFLENDTXT)

Specifies where the 'More...' and 'Bottom' text is retrieved from when displaying a subfile. The 'More...' and 'Bottom' text is displayed in a subfile when the SFLEND(*MORE) DDS keyword is specified on the subfile control record.

*MSG Use the 'More...' and 'Bottom' text retrieved from messages CPX6AB1 and CPX6AB2 which exist in the current active language of the system when the file is opened.

*FILE Use the 'More...' and 'Bottom' text that is stored in the file during file creation. This text was retrieved from messages CPX6AB1 and CPX6AB2 which exist in the active language of the system when the file was created.

User specified DBCS data (IGCDTA)

Specifies, for program-described files, whether the file processes double-byte character set (DBCS) data. Specifies, for externally described files, the DBCS attributes of the file.

For program-described files:

*NO The file does not process double-byte character set (DBCS) data.

*YES The file processes double-byte character set (DBCS) data.

For DDS files:

*NO The only DBCS attributes of the file are those defined in the DDS.

*YES DBCS attributes in addition to those defined in the DDS, include (1) putting the DDS keyword for alternative data type (IGCALTTYP) into effect, and (2) identifying DBCS attributes of fields or messages not identified in the DDS.
**DBCS extension characters (IGCEXNCHR)**

Specifies whether the system processes double-byte character set (DBCS) extension characters.

*YES  The system processes DBCS extension characters.

*NO   The system does not process DBCS extension characters; it displays them as undefined characters.

**Maximum file wait time (WAITFILE)**

Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

More information on this parameter is in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter, Appendix A.

**Note:** An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

This parameter overrides the wait time specified in the device file, in the program, or in other called OVRDSPF commands.

*IMMED  The program does not wait. Immediate allocation of file resources is required.

*CLS   The default wait time specified in the class description is used as the wait time for the allocation of the file resources.

**integer**

Specify the number of seconds that the program waits for the allocation of the file resources. Valid values range from 1 through 32767 seconds.

**Maximum record wait time (WAITRCD)**

Specifies the number of seconds the program waits for the completion of a read-from-invited-program-devices operation to a multiple device file in an HLL program. More information on how to determine when a file is considered a multiple device file is in the appropriate HLL reference manual. The program performing the read operation waits for input from all invited devices currently acquired by the file. If a record is not returned from an invited device within the specified amount of time, a notify message is sent to the program. This parameter has no effect on an input operation directed to one device.

This parameter is also used to specify the number of seconds that a CL program waits to complete a WAIT command. If a record is not returned from any of the devices that should return a record, the CL program is sent an escape message.

This parameter overrides the wait record value specified in the device file, in the program, or in other called OVRDSPF commands.
*NOMAX
   There is no limit on the amount of time the program waits for completion of a read-from-invited-program-device operation for the file.

*IMMED
   The program does not wait. If a record is not available when the read-from-invited-devices operation is done, a notify message is sent to the program.

1-32767
   Specify the number of seconds that the program waits for the completion of the read-from-invited-program-device operations.

Record format level check (LVLCHK)
Specifies whether the level identifiers of the record formats in this device file are checked when the file is opened by a program. For this check, which is done when the file is opened, the system compares the record format identifiers of each record format used by the program with the corresponding identifiers in the device file. Because the same record format name can exist in more than one file, each record format is given a unique internal system identifier when the format is created.

Level checking cannot be done unless the program contains the record format identifiers. This command cannot override level checking from *NO to *YES.

*NO   The level identifiers are not checked when the file is opened.

Secure from other overrides (SECURE)
Specifies whether this file is safe from the effects of previous call level file override commands.

*NO   This file is not protected from other file overrides. Its values can be overridden by the effects of previous call level file override commands.

*YES  This file is protected from the effects of previous call level file override commands.

Override scope (OVRSCOPE)
Specifies the extent of influence (scope) of the override.

*ACTGRPDFN
   The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

*CALLLVL
   The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

*JOB   The scope of the override is the job in which the override occurs.
Data queue (DTAQ)

Specifies the data queue that receives an entry from the system when a data-available event is signaled from an invited display device. The data queue need not exist when the display file is created since the name specified on this parameter is not evaluated until the file is used. More information on the data queue function is in the CL information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Single values

*NONE
   No data queue is specified.

Qualifier 1: Data queue

name  Specify the name of the data queue on which entries are placed.

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 data queue. If no library is specified as the current library, QGPL is used.

name  Specify the library where the data queue is located.

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the Database information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

This parameter overrides the value specified in the device file, in the program, or in other called OVRDSPF commands.

*NO  The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

*YES  The same ODP is shared with each program in the job that also specifies *YES when it opens the file.

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

*ACTGRPDFN  The scope of the open operation is determined by the activation group of the program that called the OVRDSPF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.
**JOB**  The scope of the open operation is the job in which the open operation occurs.

---

**Examples**

OVRDSPF FILE(DISPLAY75) WAITFILE(30)

This command overrides the file wait time value specified in the DISPLAY75 device file description, in the program, or in other called OVRDSPF commands. The program in which this command occurs waits up to 30 seconds (if necessary) to allocate the required file resources to the file named DISPLAY75.

---

**Error messages**

*ESCAPE Messages*

CPF1892

Function &1 not allowed.
Override ICF Pgm Device Entry (OVRICFDEVE)

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

The Override with ICF Program Device Entry (OVRICFDEVE) command can be used to temporarily add the program device entry and the remote location name to the intersystem communications function (ICF) file, or to override a program device entry with the specified remote location name and attributes for an ICF file.

**Note:** Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.


### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGMDEV</td>
<td>Program device</td>
<td>Character value</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>RMTLOCNAME</td>
<td>Remote location</td>
<td>Communications name, *REQUESTER</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>DEV</td>
<td>Device</td>
<td>Name, *LOC</td>
<td>Optional</td>
</tr>
<tr>
<td>LCLOCMNAME</td>
<td>Local location</td>
<td>Communications name, *LOC, *NETATR</td>
<td>Optional</td>
</tr>
<tr>
<td>MODE</td>
<td>Mode</td>
<td>Communications name, *NETATR</td>
<td>Optional</td>
</tr>
<tr>
<td>RMTNETID</td>
<td>Remote network identifier</td>
<td>Communications name, *LOC, *NETATR, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTSLT</td>
<td>Format select</td>
<td>*PGM, *RECID, *RMTFMT</td>
<td>Optional</td>
</tr>
<tr>
<td>APPID</td>
<td>Application identifier</td>
<td>Name, *DEVD, *USER</td>
<td>Optional</td>
</tr>
<tr>
<td>BATCH</td>
<td>Batch activity</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>HOST</td>
<td>Host type</td>
<td>*DEVD, *CICS, *IMS, *IMSRTR</td>
<td>Optional</td>
</tr>
<tr>
<td>ENDSSNHOST</td>
<td>End session with host</td>
<td>*RSHUTD, *TERMSELF</td>
<td>Optional</td>
</tr>
<tr>
<td>SPCHOSTAPP</td>
<td>Special host application</td>
<td>*DEVD, *NONE, *FLASH</td>
<td>Optional</td>
</tr>
<tr>
<td>INZSELF</td>
<td>Initialize self</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>HDRPROC</td>
<td>Header processing</td>
<td>*SYS, *USER</td>
<td>Optional</td>
</tr>
<tr>
<td>MSGPTC</td>
<td>Message protection</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>EMLDEV</td>
<td>Emulation device</td>
<td>Single values: *NONE Other values: Element list</td>
<td>Optional</td>
</tr>
<tr>
<td>Element 1: Device type</td>
<td>3278, 3284, 3286, 3287, 3288, 3289</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 2: Data format</td>
<td>*UNIFORMAT, *FIELD, *NOFIELD, *EXTFIELD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CVNTYPE</td>
<td>Conversation type</td>
<td>*SYS, *USER, *SRCPGM</td>
<td>Optional</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 1998, 2006
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Other values: Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Blocking type</td>
<td>*SEP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Element list</td>
<td></td>
</tr>
<tr>
<td>RCDLEN</td>
<td>Record length</td>
<td>1-32767, *DEVD</td>
<td>Optional</td>
</tr>
<tr>
<td>BLKLEN</td>
<td>Block length</td>
<td>1-32767, *DEVD</td>
<td>Optional</td>
</tr>
<tr>
<td>TRNSPY</td>
<td>Transmit in transparent mode</td>
<td>*DEVD, *NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>DTACPR</td>
<td>Data compression</td>
<td>*DEVD, *NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>TRUNC</td>
<td>Truncate trailing blanks</td>
<td>*DEVD, *NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OVRFLWDTA</td>
<td>Overflow data</td>
<td>*DISCARD, *RETAIN</td>
<td>Optional</td>
</tr>
<tr>
<td>GRPSEP</td>
<td>Group separator type</td>
<td>*DEVD, *EOT, *DEV3740, *OFCSYS</td>
<td>Optional</td>
</tr>
<tr>
<td>RMTBSCF</td>
<td>Remote BSCEL</td>
<td>*DEVD, *NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>INLCNN</td>
<td>Initial connection</td>
<td>*CTLD, *DIAL, *ANS</td>
<td>Optional</td>
</tr>
<tr>
<td>SECURE</td>
<td>Secure from other overrides</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OVRSCPOE</td>
<td>Override scope</td>
<td>*ACTGRPDFN, *CALLLVL, *JOB</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**Program device (PGMDEV)**

Specifies the program device name for an ICF file whose attributes are being overridden. The total number of devices that may be added to an ICF file is determined by the MAXPGMDEV parameter on the Create ICF File (CRTICFF) command or the Change ICF File (CHGICFF) command.

This is a required parameter.

*character-value*

Specify the name of the ICF program device entry with which the program communicates. This name is used in device-specific input/output operations to identify the program device and the session attributes. The program device name must be unique, although the same remote location name may be specified more than once. This allows more than one session to be at the same remote location, or to have different attribute values for each session at the same remote location. This program device name must be unique throughout the entries for the ICF file. If an override command is entered a second time for the same program device, then both (according to the override process rules) define the same program device entry.

**Note:** Refer to the APPC Programming book, SC41-5443 for information on how the system uses the RMTLOCNAME, DEV, LCLLOCNAME, and RMTNETID parameters to select an APPC device description.

**Remote location (RMTLOCNAME)**

Specifies the remote location name with which your program communicates. A remote location must be specified using the Add ICF Program Device Entry (ADDICFDEVE) command or an applied program device override. If a remote location is not given, an escape message is sent when the program device is acquired.
communications-name
Specify the name of the remote location with which your program communicates. The remote location does not need to exist at the time this command is run but must exist, either configured on the system or in the advanced peer-to-peer networking (APPN) network, for this remote location at the time the program acquires the program device. A given remote location may be added many times using different program device names. When running, however, only one program device name associated with each asynchronous (ASYNC), binary synchronous communications equivalence link (BSCEL), or system network architecture upline facility (SNUF) remote location may be acquired by the file at any one time. For each remote advanced program-to-program communication (APPC) location, more than one associated program device name may be acquired by the file at one time. For each SNUF remote location, there may be many devices. The system determines which device to use unless a device is specified on the Device (DEV) parameter.

*REQUESTER
The name used to refer to the communications device through which the program was started is used. The session that is assigned when the program device is acquired is the same session on which the Program Start request was received. If the program is not started as a result of a Program Start request, the acquire of the program device fails. The target program always uses *REQUESTER as the remote location name in the ICF file to connect to the session that the source program uses to send the Program Start request.

*REQUESTER is valid only for a target communications job. If *REQUESTER is specified in any other type of job, an escape message is sent when the program device is acquired. When *REQUESTER is used in an acquire operation, the following parameters are ignored:

- DEV
- LCLLOCNAME
- MODE
- RMTNETID

---

**Communication type (CMNTYPE)**

Specifies the communications type of the ICF device. This parameter is used only for prompting purposes; it is ignored when the command is run. The value specified for this parameter determines the subset of other parameters that are displayed (prompted) for the user.

*ALL  All parameters appear in the interactive prompt.

*APPC  Only the APPC parameters appear in the interactive prompt.

*ASYNC  Only the asynchronous parameters appear in the interactive prompt.

*BSCEL  Only the BSCEL parameters appear in the interactive prompt.

*FINANCE  Only the FINANCE parameters appear in the interactive prompt.

*RETAIL  Only the RETAIL parameters appear in the interactive prompt.

*INTRA  Only the INTRA parameters appear in the interactive prompt.
*SNUF  
Only the SNUF parameters appear in the interactive prompt.

Device (DEV)

Specifies the communications device that is used in the remote location. This parameter applies to all communications types, but should be specified only for the APPC, INTRA, and SNUF communications types. If the device is not valid for the remote location, an escape message is sent when the program device is acquired. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device and this parameter is not overridden, DEV(*LOC) is used.

*LOC  The device associated with the remote location is used. If several devices can be associated with the remote location, the system determines which device is used.

name  Specify the name of the device with which your program communicates. The device name applies to all communications types, but it should be specified only for the APPC and SNUF communications types if you want to indicate a specific device for the remote location. If the device name is not valid for the remote location, an escape message is sent when the program device is acquired.

Local location (LCLLOCNAME)

Specifies the name of your location. This parameter applies to the APPC communications type only. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device and this parameter is not overridden, LCLLOCNAME(*LOC) is used.

*LOC  The local location name associated with the remote location name is used.

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

communications-name  Specify the name of your location. The local location name is specified only in APPC if you want to indicate a specific local location name for the remote location. If the local location name is not valid for the remote location or remote location and device, an escape message is sent when the program device is acquired.

Mode (MODE)

Specifies the mode name used. This parameter applies only to the APPC communications type. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device and this parameter is not overridden, MODE(*NETATR) is used.

*NETATR  The mode in the network attributes is used.

*BLANK.  The mode name consisting of 8 blanks characters is used.
Specify a mode name for the APPC communications device. If the mode is not valid for any combination of remote location device, local location, and remote network ID, an escape message is sent when the program device is acquired.

**Remote network identifier (RMTNETID)**

Specifies the remote network ID that is used with the remote location. This parameter applies to the APPC communications type only. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device and this parameter is not overridden, RMTNETID(*LOC) is used.

- **LOC** Any remote network ID for the remote location may be used. If several remote network IDs are associated with the remote location, the system automatically selects the remote network ID.
- **NETATR** The remote network identifier specified in the network attributes is used.
- **NONE** No remote network identifier is used.

Specify a remote network ID for the APPC communications device.

**Format select (FMTSLT)**

Specifies the record format selection that is used for input operations. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device, and this parameter is not overridden, FMTSLT(*PGM) is used.

- **PGM** The program determines record format selections. If an input (read) operation with a record format name is specified, that format is always selected. If a record format is not specified for the input operation, the default format (the first record format in the file) is always selected.

  This also means that if there are any record identification keywords specified in the data description specifications (DDS) for the file, or if any remote formats are received, they are not taken into consideration when the record is selected.

- **RECID** The record identification keywords specified in the DDS for the file are used to do record selection. If there are no record identification keywords in the file, an error message is sent, the acquire operation of the program device ends, and the device is not acquired.

- **RMTFMT** The remote format names received from the sending system are used to do record selection. If the device is not an APPC device and *RMTFMT is specified, a run time error occurs at the time the program device is acquired.
Application identifier (APPID)

Specifies (in characters) the VTAM identifier of the Customer Information Control System for Virtual Storage (CICS/VS) or Information Management System/Virtual Storage (IMS/VS) host subsystem sent with the sign-on message. This parameter applies to SNUF devices only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, APPID(*DEVD) is used.

*DEVD
The application identifier specified in the device description is used.

*USER
The application program can send messages or a sign-on to the host. This is valid only when using the 3270 program interface.

name Specify the application identifier that is sent with the sign-on message.

Batch activity (BATCH)

Specifies, for CICS/VS and IMS/VS, whether this session is used for batch jobs. This parameter applies to SNUF, Retail, and INTRA devices only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, BATCH(*NO) is used.

*NO Batch jobs do not occur.

*YES Batch jobs occur and SNUF does not assemble physical records into logical records. If *YES is specified, *NO must also be specified on the Message protection (MSGPTC) parameter.

Host type (HOST)

Specifies the host or remote subsystem with which this session communicates. This parameter applies to SNUF devices only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, HOST(*DEVD) is used.

*DEVD
The host system specified in the device description is used.

*CICS The session communicates with CICS/VS.

*IMS The session communicates with IMS/VS.

*IMSRTR The session communicates with IMS/VS using the ready-to-receive option.

End session with host (ENDSSNHOST)

Specifies how the SNA upline facility (SNUF) ends the session with the host.

*RSHUTD SNUF sends a request for a turn off command to the host.

*TERMSELF SNUF sends an end-session command to the host. This value may have to be used if the value *RSHUTD fails to end a session with a non-IBM host.
Special host application (SPCHOSTAPP)

Specifies whether SNUF customizes support for special host applications outside the CICS or IMS application layer.

*DEVD  The special host application specified in the device description is used.

*NONE  SNUF does not customize support for special host applications.

*FLASH  SNUF customizes support for the Federal Reserve Flash application.

Initialize self (INZSELF)

Specifies whether a formatted INIT-SELF is built in place of the unformatted sign-on normally sent by SNUF to the host.

*NO  The unformatted default sign-on provided by SNUF is used.

*YES  The formatted INIT-SELF provided by SNUF is used.

Header processing (HDRPROC)

Specifies, for both Customer Information Control System for Virtual Storage (CICS/VS) and Information Management System for Virtual Storage (IMS/VS), whether received function management headers are passed to the application program. This parameter applies to the SNA upline facility (SNUF) communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, HDRPROC(*SYS) is used.

*SYS  SNUF removes function management headers before passing data to the program.

*USER  Function management headers are passed.

Message protection (MSGPTC)

Specifies, for both CICS/VS and IMS/VS, whether message protection is used for this session. This parameter applies to the SNUF communications type only. If the ADDICFDEVE command is not run for the specified program device, MSGPTC(*YES) is used.

*YES  Message protection is used. SNUF saves messages until they are responded to and tries to resynchronize if errors occur. *YES is valid only when *NO is also specified in the Batch activity (BATCH) parameter.

*NO  Message protection is not used.
**Emulation device (EMLDEV)**

Specifies that this program device entry is used to send and receive 3270 data streams. The emulation device parameter consists of an emulation device type and an emulation device data format. The emulation device data format specifies the format of the type 3270 data stream being sent or received. A 20- or 32-byte common header that contains type 3270 command and data flow information is located at the start of the I/O buffer that is sending or receiving the type 3270 data stream. This parameter applies only to SNUF communications. This parameter can be specified as a list of two values (elements) or as a single value (*NONE).

**Single values**

*NONE

This program device entry is not used to send and receive 3270 data streams.

**Element 1: Device type**

3278  The data stream is for a 3278, 3277, or 3279 display device.

3284  The data stream is for a 3284 printer device.

3286  The data stream is for a 3286 printer device.

3287  The data stream is for a 3287 printer device.

3288  The data stream is for a 3288 printer device.

3289  The data stream is for a 3289 printer device.

**Element 2: Data format**

*UNFORMAT

An unformatted 3270 data stream is sent or received. The user application program must translate the data stream into a display or printer image.

*FIELD

A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display or printer image that contains field definitions. The field definitions indicate the location and characteristics of the fields. *FIELD is valid only if *NO is specified on the BATCH parameter.

*NIFIELD

A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display or printer image without field definitions. *NIFIELD is valid only if *NO is specified on the BATCH parameter.

*EXTFIELD

A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display image followed by field definitions. The field definitions indicate the location and characteristics of fields. *EXTFIELD is valid only if *NO is specified on the BATCH parameter and 3278 is specified as the emulation device type.

**Conversation type (CNVTYPE)**

Specifies the conversation type for which the application program is designed. This parameter is valid for advanced program-to-program communications (APPC) communications types only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, CNVTYPE(*SYS) is used. More information on the APPC communications type can be found in the APPC Programming book, SC41-5443.
**SYS** The APPC mapped conversation support for the LU 6.2 architecture is used.

**USER**
The APPC basic conversation support for the LU 6.2 architecture is used.

**SRCPGM**
The target program accepts the conversation type specified by the source program.

---

**Blocking type (BLOCK)**

Specifies whether the system or the user controls whether records are combined into blocks when they are sent. This parameter is for the BSCEL communications type. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, BLOCK(*DEVD) is used.

With this parameter, you may specify one of the following conditions of record formatting:

- No blocking or deblocking: The record format described in the DDS is the format for both the record and the block.
- User blocking or deblocking: Specify the BSCEL controls needed to describe the record format of the system.
- System blocking with record separator characters: Specify the record separator character used by the system to determine record boundaries in the block.
- System blocking of fixed-length records: The system uses fixed-length records, and blocks or deblocks records accordingly.

If you specify a parameter value other than *NONE or *USER, records are blocked as required by the system for output, and are deblocked on input.

**Single values**

**DEVD**
The block option in the device description is used.

**NONE**
Blocking or deblocking is not done by the system.

**ITB**
The records are blocked or deblocked based on the location of an intermediate text block (ITB) control character. For input files, a record is delimited by locating the next intermediate text block character. An end-of-text or end-of-transmission block character is used as an intermediate text block character to delimit a block. For output files, an ITB character is added after the record. If it is the last character of the block, the ITB is replaced by an end-of-text or end-of-transmission block character.

**IRS**
The records are blocked or deblocked based on the location of an interrecord separator (IRS) character. For input files, a record is delimited by locating the next IRS character. For output files, an IRS character is added after the record.

**NOSEP**
No record separator character is contained in the block that is sent to or received from the device. The system blocks and deblocks the records using a fixed-length record, as specified in the DDS format specifications.

**USER**
The program gives all control characters, including record separator characters, BSCEL framing characters, transparency characters, and so forth, necessary to send records.

**Element 1: Blocking type**
*SEP  The records are blocked or deblocked based on the location of a user-specified record separator character. For input files, a record is delimited by locating the next record separator character. For output files, a record separator character is added after the record.

Element 2: Record separator, if *SEP

X'1E'  The record separator character X'1E' is used.

**hexadecimal-value**

Specify a record separator character that is unique and has a length of 1 byte. The record separator character may be specified as 2 hexadecimal characters, as in BLOCK(*SEP X'FD'), or the character may be specified as a single character, as in BLOCK(*SEP @).

---

**Record length (RCDLEN)**

Specifies the maximum record length (in bytes) for data sent and received. This parameter applies to the SNUF and BSCEL communications types only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, RCDLEN(*DEVD) is used.

*DEVD  The record length specified in the device description is used. If a record is longer than the specified record length, a run time error occurs at the time the record is sent or received.

1-32767  Specify the maximum record length to use for this device file. The value must be at least the size of the largest record sent. If a record is longer than the specified record length, a run time error occurs when the record is sent or received. Valid values range from 1 through 32767 for SNUF communication. For BSCEL communication, the maximum record length is 8192 bytes.

---

**Block length (BLKLEN)**

Specifies the maximum block length (in bytes) for data sent. This parameter applies to the BSCEL and SNUF communications types only. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device, or is not overridden, this parameter defaults to *DEVD.

*DEVD  The block length specified in the device description is used.

1-32767  Specify the maximum block length of records sent when this device file is used. The value must be at least the size of the largest record sent. Valid values range from 1 to 32767 for SNUF communication. For BSCEL communication, the maximum block length is 8192 bytes.

---

**Transmit in transparent mode (TRNSPY)**

Specifies whether text is sent in transparent text mode. Transparent text mode allows all 256 extended binary-coded decimal interchange code (EBCDIC) character codes to be sent. Use this feature when sending packed or binary data fields. This parameter is for the BSCEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, TRNSPY(*DEVD) is used.
*DEVD
   The text transparency option specified in the device description is used.

*NO
   Text transparency is not used.

*YES
   Text transparency is used, which allows all 256 EBCDIC character codes to be sent. *YES is valid only when *NONE, *NOSEP, or *USER is specified in the Blocking type (BLOCK) parameter.

Data compression (DTACPR)
Specifies whether blanks in BSCEL data are compressed for output and decompressed for input. *YES cannot be specified if *YES is specified in the Transmit in transparent mode (TRNSPY) parameter. This parameter is for the BSCEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, DTACPR(*DEVD) is used.

*DEVD
   The data compression option specified in the device description is used.

*NO
   No data compression or decompression occurs.

*YES
   Data is compressed for output and decompressed for input.

Truncate trailing blanks (TRUNC)
Specifies whether trailing blanks are removed from output records. *YES cannot be specified if *NOSEP is specified in the Blocking type (BLOCK) parameter. If *YES is specified and *YES is also specified in the Data compression (DTACPR) parameter, then truncation is ignored. This parameter is for BSCEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, DTACPR(*DEVD) is used.

*DEVD
   The trailing blanks specified in the device description are used.

*NO
   Trailing blanks are not removed from output records.

*YES
   Trailing blanks are removed from output records.

Overflow data (OVRFLWDTA)
Specifies whether overflow data is discarded or kept.

*DISCARD
   Overflow data is not kept.

*RETAIIN
   Overflow data is kept.
Group separator type (GRPSEP)

Specifies a separator for groups of data, such as data sets and documents. This parameter is for the BSCEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, GRPSEP(*DEVD) is used.

*DEVD  
The group separator option specified in the device description is used.

*EOT  
The BSCEL control character EOT (end-of-transmission) is used as a data group separator.

*DEV3740  
A null record (STXETX) is used as a data group separator.

*OFCSYS  
A block sent with the BSCEL control character ETX (end-of-information) is used as a data group separator.

Remote BSCEL (RMTBSCEL)

Specifies whether the type of BSCEL session is with a BSCEL system. This parameter applies to the BSCEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, RMTBSCEL(*DEVD) is used.

*DEVD  
The option for BSCEL specified in the device description is used.

*NO  
An attribute of *NO indicates the remote system cannot recognize BSCEL commands or messages. In most cases, *NO is used when communicating with remote systems such as a 3741 Data Entry Station, an Office System 6, a 5230 Data Collection System, or a System/38.

*YES  
The remote system can recognize the BSCEL transaction starting commands, transaction ending commands, and online messages. In most cases, *YES indicates that the remote system is either another iSeries computer, AS/400 system, a System/38, a System/36, or a System/34 with BSCEL support.

Initial connection (INLCNN)

Specifies the method of making a connection on the line for the session being acquired. This parameter is valid for the BSCEL communications type only. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device, or is not overridden, this parameter defaults to *CTLD.

*CTLD  
The initial connection option specified in the controller description is used.

*DIAL  
The local system starts the call.

*ANS  
The remote system starts the call, and the local system answers the call.
Secure from other overrides (SECURE)

Specifies whether this program device is protected from the effects of override commands issued in earlier programs.

*NO  This program device override is not protected from other program device overrides. Its values can be overridden by the effects of any program device override commands issued in earlier programs.

*YES  This program device override is protected from the effects of any program device override commands issued in earlier programs.

Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.

*ACTGRPDEFN
The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

*CALLLVL
The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

*JOB  The scope of the override is the job in which the override occurs.

Examples

Example 1: Overriding the Device Entry with the Record Format Selection Attributes
OVRICFDEVE PGMDEV(BSCEL2) RMTLOCNAME(BSCNYC)
    FMTSLT(*RECID)

This command overrides the program device named BSCEL2 with a corresponding remote location named BSCNYC for any ICF file associated with the job. The program device is overridden with the attributes of FMTSLT(*RECID).

Example 2: Overriding the Device Entry with the Record Format Selection and the Conversation Type Attributes
OVRICFDEVE PGMDEV(APPC1) RMTLOCNAME(*REQUESTER)
    FMTSLT(*RMTFMT) CNVTYPE(*SYS)

This command overrides the program device entry named APPC1 with a remote location name of *REQUESTER. The program device entry is overridden with the FMTSLT(*RMTFMT) and CNVTYPE(*SYS) attributes.

Example 3: Overriding an Entry for Associated ICF Files
OVRICFDEVE PGMDEV(JOE) RMTLOCNAME(LU0MPLS)

This command overrides the program device entry named JOE with a remote location named LU0MPLS for any ICF file associated with the job.
Example 4: Specifying the Communications Device

OVRICFDEV   PGMDEV(APPC) RMTLOCNAME(APPCMPLS) DEV(MPLSLINE2)

This command overrides the program device entry named APPC with a remote location named APPCMPLS using device MPLSLINE2.

---

Error messages

*ESCAPE Messages

CPF180C
   Function &1 not allowed.

CPF1892
   Function &1 not allowed.
Override ICF File (OVRICFF)

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

The Override with Intersystem Communications Function File (OVRICFF) command overrides the file named in the program and overrides certain parameters of the file being processed. Parameters overridden by this command can be specified in the file description, in the program, or in other file override commands that are run later.

If a file named in the program is being overridden, the name of that file is specified in the FILE parameter and the name of the overriding file (the file being processed) is specified in the TOFILE parameter.

This command can also specify parameters to override values contained in the file description of the overriding file. If the file named in the program is not being replaced but certain parameters of the file are being overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters being overridden are then specified by the other parameters of the OVRICFF command. Parameters that are not specified do not affect parameters specified in the file description, in the program, or in other override commands run later.


Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>File being overridden</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>TOFILE</td>
<td>Overriding to ICF file</td>
<td>Single values: *FILE</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Overriding to ICF file</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>ACQPGMDEV</td>
<td>Acquire program device</td>
<td>Element list</td>
<td>Optional, Positional 3</td>
</tr>
<tr>
<td></td>
<td>Element 1: Program device to acquire</td>
<td>Character value, *NONE</td>
<td></td>
</tr>
<tr>
<td>MAXRCDLEN</td>
<td>Maximum record length</td>
<td>Integer, *CALC</td>
<td>Optional</td>
</tr>
<tr>
<td>WAITFILE</td>
<td>Maximum file wait time</td>
<td>Integer, *IMMED, *CLS</td>
<td>Optional</td>
</tr>
<tr>
<td>WAITRCD</td>
<td>Maximum record wait time</td>
<td>1-32767, *NOMAX, *IMMED</td>
<td>Optional</td>
</tr>
<tr>
<td>LVLCHK</td>
<td>Record format level check</td>
<td>*NO</td>
<td>Optional</td>
</tr>
<tr>
<td>SECURE</td>
<td>Secure from other overrides</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Parameters

© Copyright IBM Corp. 1998, 2006
### File being overridden (FILE)

Specifies the file to which this override command is applied. The specified file must be an intersystem communications function (ICF) file when *FILE is specified in the **Overriding to ICF file (TOFILE)** parameter. Otherwise, any device file or database file name can be specified.

This is a required parameter.

**name** Specify the name of the ICF file.

### Overriding to ICF file (TOFILE)

Specifies the qualified name of the ICF file (up to 10 characters) that is used instead of the file specified in the **File being overridden (FILE)** parameter or, if *FILE is specified, indicates that certain attributes are overridden by parameters in this command. The parameters on this command override the other values in the ICF file or in the program.

**Single values**

**FILE** Some parameters of the ICF file named in the **File being overridden (FILE)** parameter are overridden by values specified in this command.

**Qualifier 1: Overriding to ICF file**

**name** Specify the name of the ICF file that is used instead of the overridden 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 ICF file description. If no library is specified as the current library for the job, QGPL is used.

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

### Acquire program device (ACQPGMDEV)

Specifies which program device is acquired by the file when the file is opened. This parameter overrides the value in the ICF file, in the program, or in the other OVRICFF commands run later.
**NONE**

The file is opened without any devices acquired. All devices used with this file must be explicitly acquired before input/output can be directed to them.

**character-value**

Specify the name of the program device that is acquired when the file is opened. The name should be specified on the Add ICF Program Device Entry (ADDICFDEVE) command or the override ICF program device entry (OVRICFDEVE) command as a program device name before the file is opened.

---

**Maximum record length (MAXRCDLEN)**

Specifies the maximum record length used when the file is opened. This parameter overrides the value in the ICF file, in the program, or in the other OVRICFF commands run later.

**CALC**

The value calculated in the file is used when the file is opened.

**integer**

Specify the record length (in characters) that is used when the file is opened. Valid values range from 1 through 32767. If the record length is less than the calculated value in the file, the calculated value is used.

---

**Maximum file wait time (WAITFILE)**

Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

**IMMED**

The program does not wait. Immediate allocation of file resources is required.

**CLS**

The default wait time specified in the class description is used as the wait time for the allocation of the file resources.

**integer**

Specify the number of seconds that the program waits for the allocation of the file resources. Valid values range from 1 through 32767.

---

**Maximum record wait time (WAITRCD)**

Specifies the number of seconds the program waits for the completion of a read-from-invited-devices operation to a multiple device file in a high-level language program. Refer to the high-level language reference manual to determine when a file is treated as a multiple device file. The program performing the read operation waits for the input form all invited devices currently accessing the file. If a record is not returned from any of the invited program devices in the specified amount of time, a notify message is sent to the program. This parameter has no effect on an input operation directed to a single device.

**NOMAX**

There is no limit on the time the system waits for the completion of the operation.
*IMMED

The program does not wait. If a record is not available when the read-from-invited-devices operation is done, a notify message is sent to the program.

1-32767

Specify the number of seconds that the program waits for the completion of the read-from-invited-program-devices operation.

---

**Record format level check (LVLCHK)**

Specifies whether the level identifiers of the record formats in this device file are checked when the file is opened by a program. While the file is being opened, the system verifies the level identifiers and compares the record format identifiers of each record format used by the program with the same identifiers in the device file. Because the same record format name can exist in more than one file, each record format is given a unique internal system identifier when the format is created.

**Note:** This command cannot override level checking from *NO to *YES.

*NO  The level identifiers of the record formats are not checked when the file is opened.

---

**Secure from other overrides (SECURE)**

Specifies whether this file is protected from the effects of previous file override commands.

*NO  This file is not protected from other file overrides. Its values can be overridden by the effects of any file override commands started earlier.

*YES  This file is protected from the effects of any file override commands started earlier.

---

**Override scope (OVRSCOPE)**

Specifies the extent of influence (scope) of the override.

*ACTGRPDFN

The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

*CALLEVL

The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

*JOB  The scope of the override is the job in which the override occurs.
Data queue (DTAQ)

Specifies the data queue on which entries are placed. The specified data queue must have a minimum length of 80 characters. The data queue need not exist when the display file is created since the name specified for this parameter is not evaluated until the file is used.

**Note:** Keyed data queues are not supported for this parameter. If a keyed data queue is specified, a run-time error will occur; but because it is not required that a data queue exist at the time the command is issued, the error will not be flagged.

More information on the data queue function is in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Single values

*NONE
  No data queue is specified.

Qualifier 1: Data queue

name Specify the name of the data queue on which entries are placed.

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 data queue. If no library is specified as the current library, QGPL is used.

name Specify the library where the data queue is located.

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

*NO
  The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

*YES
  The same ODP is shared with each program in the job that also specifies *YES when it opens the file.

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

*ACTGRPDFN
  The scope of the open operation is determined by the activation group of the program that called the OVRICFF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

*JOB
  The scope of the open operation is the job in which the open operation occurs.
Examples

OVRICFF FILE(ICFHIST) TOFILE(PRSNNL/ICFCURT)

This command overrides the file named ICFHIST to the ICF file named ICFCURT in library PRSNNL.

Error messages

*ESCAPE Messages

CPF180C
  Function &1 not allowed.

CPF1892
  Function &1 not allowed.
Override Message File (OVRMSGF)

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

The Override with Message File (OVRMSGF) command overrides a message file used in a program. The overriding message file is used (specified in the TOMSGF parameter) whenever a message is sent or retrieved and the overridden message file is specified.

The overriding message file need not contain all the messages that the overridden file contains. When a message is received or retrieved and the message identifier cannot be found in the overriding message file, the overridden message file is searched for the identifier. Overriding message files can be overridden, resulting in a chain of overrides. This chain of overrides provides a list of message files that are searched when a message is received or retrieved. Up to 30 message file overrides can be specified in a program.

Restrictions:
• In a multithreaded job, this command may only be issued from the initial thread.
• In a multithreaded job, this command will only affect message file references performed in the initial thread. Message file references performed in secondary threads will be unaffected.


Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSGF</td>
<td>Message file being overridden</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>TOMSGF</td>
<td>Overriding to message file</td>
<td>Qualified object name</td>
<td>Required, Positional 2</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Overriding to message file</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>SECURE</td>
<td>Secure from other overrides</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Message file being overridden (MSGF)

Specifies the message file being used by the program to which this override command is applied.

This is a required parameter.
**name**  Specify the name of the message file.

---

**Overriding to message file (TOMSGF)**

Specifies the message file that is used instead of the message file specified in the **Message file being overridden (MSGF)** parameter; or, if the names are the same, specifies that the value specified in the **Secure from other overrides (SECURE)** parameter is used for the message file.

This is a required parameter.

**Qualifier 1: Overriding to message file**

**name**  Specify the name of the message file that is used instead of the overridden message 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 message file. If no library is specified as the current library for the job, QGPL is used.

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

---

**Secure from other overrides (SECURE)**

Specifies whether this file is secured from the effects of message file override commands used in earlier calls. If this parameter is not specified, processing occurs as if *NO* had been specified.

- **NO**  This message file is not protected from other file overrides. Its values can be overridden by the effects of any message file overrides used in earlier calls.
- **YES**  This message file is protected from the effects of any message file overrides used in earlier calls.

---

**Examples**

OVRMSGF  MSGF (WSUSRMSG)  TOMSGF (ORDENTMSGD)

This override command causes the defaults for messages stored in ORDENTMSGD to be used instead of defaults stored in WSUSRMSG (which contains messages designed for work station users). As a result of this command, the messages received by the order entry users are tailored to their own environment.

---

**Error messages**

- **ESCAPE Messages**
  - **CPF180C**  Function &1 not allowed.
Override with Printer File (OVRPRTF)

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

The Override with Printer File (OVRPRTF) command is used to (1) override (replace) the file named in the program, (2) override certain parameters of a file that are used by the program, or (3) override the file named in the program and override certain parameters of the file processed. Parameters overridden by this command are specified in the file description, in the program, or in other file override commands that run in the following command.

If a file named in the program is overridden, the name of that file is specified in the FILE parameter and the name of the overriding file (the file processed) is specified in the TOFILE parameter. The OVRPRTF command also specifies parameters to override values contained in the file description of the overriding file. If the file named in the program is not replaced but certain parameters of the file are overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters overridden are then specified by the other parameters of the OVRPRTF command. Parameters not specified do not affect parameters specified in the file description, in the program, or in other file override commands run later.

Restrictions:
- In a multithreaded job, this command may only be issued from the initial thread.
- In a multithreaded job, only Activation Group or Job scoped overrides will affect opens performed in a secondary thread.

More information on overriding files is in the Basic Printing information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>File being overridden</td>
<td>Name, *PRTF</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>TOFILE</td>
<td>Overriding to printer file</td>
<td>Single values: *FILE</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Overriding to printer file</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>DEV</td>
<td>Device</td>
<td>Element list</td>
<td>Optional, Positional 3</td>
</tr>
<tr>
<td></td>
<td>Element 1: Printer</td>
<td>Name, *SYSVAL, *JOB</td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------</td>
<td>------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>PAGESIZE</td>
<td>Page size</td>
<td>*Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Page length</td>
<td>0.001-2.55.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Page width</td>
<td>0.001-378.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 3: Measurement method</td>
<td>*ROWCOL, *UOM</td>
<td></td>
</tr>
<tr>
<td>LPI</td>
<td>Lines per inch</td>
<td>3.0, 4.0, 6.0, 7.5, 7.5, 8.0, 9.0, 12.0</td>
<td>Optional</td>
</tr>
<tr>
<td>CPI</td>
<td>Characters per inch</td>
<td>5.0, 10.0, 12.0, 13.3, 15.0, 16.7, 16.7, 18.0, 20.0</td>
<td>Optional</td>
</tr>
<tr>
<td>FRONTMGN</td>
<td>Front margin</td>
<td>Single values: *DEVD</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Other values: *Element list</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Offset down</td>
<td>0.0-57.79</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Offset across</td>
<td>0.0-57.79</td>
<td></td>
</tr>
<tr>
<td>BACKMGN</td>
<td>Back margin</td>
<td>Single values: *FRONTMGN, *DEVD</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Other values: *Element list</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Offset down</td>
<td>0.0-57.79</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Offset across</td>
<td>0.0-57.79</td>
<td></td>
</tr>
<tr>
<td>OVRFLW</td>
<td>Overflow line number</td>
<td>1-255</td>
<td>Optional</td>
</tr>
<tr>
<td>FOLD</td>
<td>Fold records</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>RPLUNPRT</td>
<td>Unprintable character action</td>
<td>Single values: *NO</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Other values: *Element list</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Replace character</td>
<td>*YES</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 2: Replacement character</td>
<td>X’40’-X’FE’, ‘ ’</td>
<td></td>
</tr>
<tr>
<td>ALIGN</td>
<td>Align page</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>DRAWER</td>
<td>Source drawer</td>
<td>1-255, *EL, *FORMDF</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTBIN</td>
<td>Output bin</td>
<td>1-65535, *DEVD</td>
<td>Optional</td>
</tr>
<tr>
<td>FONT</td>
<td>Font</td>
<td>Single values: *CPI, *DEVD</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Other values: *Element list</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Point size</td>
<td>0.1-999.9, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>FORMFEED</td>
<td>Form feed</td>
<td>*DEVD, *CONT, *CUT, *CONT2, *AUTOCUT</td>
<td>Optional</td>
</tr>
<tr>
<td>CTLCHAR</td>
<td>Control character</td>
<td>*NONE, *FCFC, *MACHINE</td>
<td>Optional</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>CHLVAL</td>
<td>Channel values</td>
<td>Single values: *NORMAL Other values (up to 12 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td>Element 1: Channel</td>
<td>1-12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 2: Line number for channel</td>
<td>Element list</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 1: Line</td>
<td>1-255</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIDELITY</td>
<td>Fidelity</td>
<td>*CONTENT, *ABSOLUTE</td>
<td>Optional</td>
</tr>
<tr>
<td>CHRID</td>
<td>Character identifier</td>
<td>Single values: *DEVD, *SYSVAL, *JOBCCSID, *CHRIDCTL Other values: Element list</td>
<td>Optional</td>
</tr>
<tr>
<td>Element 1: Graphic character set</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 2: Code page</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECFMT</td>
<td>Decimal format</td>
<td>*FILE, *JOB</td>
<td>Optional</td>
</tr>
<tr>
<td>FNTCHRSET</td>
<td>Font character set</td>
<td>Single values: *FONT Other values: Element list</td>
<td>Optional</td>
</tr>
<tr>
<td>Element 1: Character set</td>
<td>Qualified object name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 1: Character set</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 2: Code page</td>
<td>Qualified object name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 1: Code page</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 3: Point size</td>
<td>0.1-999.9, *NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDEFNT</td>
<td>Coded font</td>
<td>Single values: *FNTCHRSET Other values: Element list</td>
<td>Optional</td>
</tr>
<tr>
<td>Element 1: Coded font</td>
<td>Qualified object name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 1: Coded font</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 2: Point size</td>
<td>0.1-999.9, *NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAGDFN</td>
<td>Page definition</td>
<td>Single values: *NONE Other values: Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td>Qualifier 1: Page definition</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORMDF</td>
<td>Form definition</td>
<td>Single values: *NONE, *DEVD Other values: Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td>Qualifier 1: Form definition</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFPCHARS</td>
<td>AFP Characters</td>
<td>Single values: *NONE Other values (up to 4 repetitions): Name</td>
<td>Optional</td>
</tr>
<tr>
<td>TBLREFCHR</td>
<td>Table Reference Characters</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>PAGRTT</td>
<td>Degree of page rotation</td>
<td>*AUTO, *DEVD, *COR, 0, 90, 180, 270</td>
<td>Optional</td>
</tr>
<tr>
<td>MULTIUP</td>
<td>Pages per side</td>
<td>1-4, 1</td>
<td>Optional</td>
</tr>
<tr>
<td>REDUCE</td>
<td>Reduce output</td>
<td>*TEXT, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>PRTTXT</td>
<td>Print text</td>
<td>Character value, *JOB, *BLANK, X”</td>
<td>Optional</td>
</tr>
<tr>
<td>JUSTIFY</td>
<td>Hardware justification</td>
<td>0, 50, 100</td>
<td>Optional</td>
</tr>
<tr>
<td>DUPLEX</td>
<td>Print on both sides</td>
<td>*NO, *YES, *TUMBLE, *FORMDF</td>
<td>Optional</td>
</tr>
<tr>
<td>UOM</td>
<td>Unit of measure</td>
<td>*INCH, *CM</td>
<td>Optional</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>FRONTOVL</td>
<td>Front side overlay</td>
<td>Single values: *NONE&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB&lt;br&gt;Element 2: Offset down 0.0-57.79, 0&lt;br&gt;Element 3: Offset across 0.0-57.79, 0</td>
<td>Optional</td>
</tr>
<tr>
<td>BACKOVL</td>
<td>Back side overlay</td>
<td>Single values: *FRONTOVL, *NONE&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB&lt;br&gt;Element 2: Offset down 0.0-57.79, 0&lt;br&gt;Element 4: Constant back *NOCONSTANT, *CONSTANT</td>
<td>Optional</td>
</tr>
<tr>
<td>CVTLINDTA</td>
<td>Convert line data</td>
<td>*NO, *YES&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB</td>
<td>Optional</td>
</tr>
<tr>
<td>IPDSPASTHR</td>
<td>IPDS pass through</td>
<td>*YES, *NO, *DEVD&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB&lt;br&gt;Element 2: Offset down 0.0-57.79, 0&lt;br&gt;Element 3: Offset across 0.0-57.79, 0</td>
<td>Optional</td>
</tr>
<tr>
<td>USRRSCLIBL</td>
<td>User resource library list</td>
<td>Single values: *DEVD, *NONE, *JOBLIBL, *CURLIB&lt;br&gt;Other values (up to 4 repetitions): *Character value&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB&lt;br&gt;Element 2: Offset down 0.0-57.79, 0&lt;br&gt;Element 3: Offset across 0.0-57.79, 0</td>
<td>Optional</td>
</tr>
<tr>
<td>CORNERSTPL</td>
<td>Corner staple</td>
<td>*NONE, *BOTRIGHT, *TOPRIGHT, *TOPLEFT, *BOTLEFT, *DEVD&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB&lt;br&gt;Element 2: Offset down 0.0-57.79, 0&lt;br&gt;Element 3: Offset across 0.0-57.79, 0&lt;br&gt;Element 4: Constant back *NOCONSTANT, *CONSTANT</td>
<td>Optional</td>
</tr>
<tr>
<td>EDGESTITCH</td>
<td>Edge stitch</td>
<td>Single values: *NONE&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Reference edge *BOT, *RIGHT, *TOP, *LEFT, *DEVD&lt;br&gt;Element 2: Reference edge offset 0.0-57.79, *DEVD&lt;br&gt;Element 3: Number of staples 1-122, *DEVD&lt;br&gt;Element 4: Staple offsets Values (up to 122 repetitions): 0.0-57.79, *DEVD</td>
<td>Optional</td>
</tr>
<tr>
<td>SADLSTITCH</td>
<td>Saddle stitch</td>
<td>Single values: *NONE&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Reference edge *TOP, *LEFT, *DEVD&lt;br&gt;Element 2: Number of staples 1-122, *DEVD&lt;br&gt;Element 3: Staple offsets Values (up to 122 repetitions): 0.0-57.79, *DEVD</td>
<td>Optional</td>
</tr>
<tr>
<td>FNTRESL</td>
<td>Font resolution for formatting</td>
<td>*DEVD, *SEARCH, 240, 300&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB</td>
<td>Optional</td>
</tr>
<tr>
<td>DFRWRT</td>
<td>Defer write</td>
<td>*YES, *NO&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB</td>
<td>Optional</td>
</tr>
<tr>
<td>SPOOL</td>
<td>Spool the data</td>
<td>*YES, *NO&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTQ</td>
<td>Output queue</td>
<td>Single values: *DEV, *JOB&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Output queue Name&lt;br&gt;Qualifier 1: Output queue Name, *LIBL, *CURLIB&lt;br&gt;Element 2: Library Name, *LIBL, *CURLIB</td>
<td>Optional</td>
</tr>
<tr>
<td>FORMTYPE</td>
<td>Form type</td>
<td>Character value, *STD&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB&lt;br&gt;Element 2: Offset down 0.0-57.79, 0&lt;br&gt;Element 3: Offset across 0.0-57.79, 0&lt;br&gt;Element 4: Constant back *NOCONSTANT, *CONSTANT</td>
<td>Optional</td>
</tr>
<tr>
<td>COPIES</td>
<td>Copies</td>
<td>1-255&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB&lt;br&gt;Element 2: Offset down 0.0-57.79, 0&lt;br&gt;Element 3: Offset across 0.0-57.79, 0&lt;br&gt;Element 4: Constant back *NOCONSTANT, *CONSTANT</td>
<td>Optional</td>
</tr>
<tr>
<td>PAGERANGE</td>
<td>Page range to print</td>
<td>Element list&lt;br&gt;Element 1: Starting page Integer, 1, *ENDPAGE&lt;br&gt;Element 2: Ending page Integer, *END&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB&lt;br&gt;Element 2: Offset down 0.0-57.79, 0&lt;br&gt;Element 3: Offset across 0.0-57.79, 0&lt;br&gt;Element 4: Constant back *NOCONSTANT, *CONSTANT</td>
<td>Optional</td>
</tr>
<tr>
<td>MAXRCDS</td>
<td>Max spooled output records</td>
<td>1-999999, *NOMAX&lt;br&gt;Other values: *Element list&lt;br&gt;Element 1: Overlay Qualified object name&lt;br&gt;Qualifier 1: Overlay Name&lt;br&gt;Qualifier 2: Library Name, *LIBL, *CURLIB&lt;br&gt;Element 2: Offset down 0.0-57.79, 0&lt;br&gt;Element 3: Offset across 0.0-57.79, 0&lt;br&gt;Element 4: Constant back *NOCONSTANT, *CONSTANT</td>
<td>Optional</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------</td>
<td>----------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>FILESEP</td>
<td>File separators</td>
<td>0-9</td>
<td>Optional</td>
</tr>
<tr>
<td>SCHEDULE</td>
<td>Spooled output schedule</td>
<td>*JOBEND, *FILEEND, *IMMED</td>
<td>Optional</td>
</tr>
<tr>
<td>HOLD</td>
<td>Hold spooled file</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>SAVE</td>
<td>Save spooled file</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTPTY</td>
<td>Output priority (on OUTQ)</td>
<td>*JOB, 1, 2, 3, 4, 5, 6, 7, 8, 9</td>
<td>Optional</td>
</tr>
<tr>
<td>USRDTA</td>
<td>User data</td>
<td>Character value, *SOURCE</td>
<td>Optional</td>
</tr>
<tr>
<td>SPLFOWN</td>
<td>Spool file owner</td>
<td>*CURUSRPRF, *JOB, *CURGRPPRF, *JOBGRPPRF</td>
<td>Optional</td>
</tr>
<tr>
<td>USRDFNOPT</td>
<td>User Defined Option</td>
<td>Single values: *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>USRDFNDA</td>
<td>User Defined Data</td>
<td>Character value, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>USRDFNOBJ</td>
<td>User Defined Object</td>
<td>Single values: *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Element 1: Object</td>
<td>Qualified object name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Qualifier 1: Object</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
</tr>
<tr>
<td>SPLFNAME</td>
<td>Spool file name</td>
<td>Name, *FILE</td>
<td>Optional</td>
</tr>
<tr>
<td>EXPDATE</td>
<td>Expiration date for file</td>
<td>Date, *NONE, *DAYS</td>
<td>Optional</td>
</tr>
<tr>
<td>DAYS</td>
<td>Days until file expires</td>
<td>1-366</td>
<td>Optional</td>
</tr>
<tr>
<td>IGCDTA</td>
<td>User specified DBCS data</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>IGCEXCHR</td>
<td>DBCS extension characters</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>IGCHRRIT</td>
<td>DBCS character rotation</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>IGCCPI</td>
<td>DBCS characters per inch</td>
<td>*CPI, *CONDENSED, 5, 6, 10</td>
<td>Optional</td>
</tr>
<tr>
<td>IGCSOSI</td>
<td>DBCS SO/SI spacing</td>
<td>*YES, *NO, *RIGHT</td>
<td>Optional</td>
</tr>
<tr>
<td>IGCCDEFNT</td>
<td>DBCS coded font</td>
<td>Single values: *SYSVAL</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Element 1: DBCS coded font</td>
<td>Qualified object name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Qualifier 1: DBCS coded font</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Element 2: Point size</td>
<td>0.1-999.9, *NONE</td>
</tr>
<tr>
<td>WAITFILE</td>
<td>Maximum file wait time</td>
<td>Integer, *IMMED, *CLS</td>
<td>Optional</td>
</tr>
<tr>
<td>LVLCHK</td>
<td>Record format level check</td>
<td>*NO</td>
<td>Optional</td>
</tr>
<tr>
<td>SECURE</td>
<td>Secure from other overrides</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OVRSCOPE</td>
<td>Override scope</td>
<td>*ACTGRPDEF, *CALLLVL, *JOB</td>
<td>Optional</td>
</tr>
<tr>
<td>SHARE</td>
<td>Share open data path</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OPNSCOPE</td>
<td>Open scope</td>
<td>*ACTGRPDEF, *JOB</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**File being overridden (FILE)**

Specifies the file being used by the program to which this override command is applied. The specified file must be a printer device file when *FILE is specified in the Overriding to printer file (TOFILE) parameter. Otherwise, any device file name or database file name is specified.

This is a required parameter.
*PRTF  The *PRTF override is applied. This override applies to all printer files being opened except for those printer files that already have specific overrides. For example, if a *PRTF override is issued at call level 3, and an override is issued for QSYSPRTR at call level 3, the *PRTF override is applied to all printer files being opened except for QSYSPRTR since there is a specific override for it.

name  Specify the names of one or more overridden files for which the overrides in the call level are applied.

Overriding to printer file (TOFILE)  
Specifies the printer file that is used instead of the file specified in the File being overridden (FILE) parameter; or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified on this OVRPRTF command override the same parameters specified in the printer file, in the program, or in other called (OVRPRTF) commands.

Single values  
*FILE  The printer device file named in the FILE parameter has some of its parameters overridden by values specified in this command.

Qualifier 1: Overriding to printer file  
name  Specify the name of the printer device file that is used instead of the overridden 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 used as the current library for the job, QGPL is used.

name  Specify the library where the file is located.

Device (DEV)  
Specifies a printer device description. For nonspooled output, this identifies the printer device used to produce the printed output. For spooled output, the file is placed on the output queue determined by the OUTQ parameter. If OUTQ(*DEV) is used, the file is placed on the output queue with the same name as the device.

*SYSVAL  The value in the system value QPRTDEV at the time the job is started is used as the printer device.

*JOB  The printer associated with the job is the printer device.

name  Specify the name of the device that is used with the printer file. The device name must already be known on the system by a device description.

Double-byte character set considerations: When printing a file that has double-byte character set (DBCS) data, specify a DBCS printer (5553, 5583).
Printer device type (DEVTYPE)

Specifies the type of data stream that is created for a printer device file. This parameter indicates whether the resulting data stream is an Intelligent Printer Data Stream (IPDS) or an SNA Character Stream (SCS).

*SCS  An SNA Character Stream (SCS) data stream is created.

Note: When using double-byte character set (DBCS) printers (the 5553 and 5583 Printers), DEVTYPE(*SCS) must be specified.

*IPDS  An Intelligent Printer Data Stream (IPDS) is created. This parameter can be specified when using an IPDS printer. If *IPDS is specified and the spooled printer file is directed to a printer other than an IPDS printer, the IPDS printer file is converted to an SCS printer file. More information is in the Basic Printing information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

*USERASCII  An ASCII data stream is placed on a spooled output queue. You are responsible for placing the entire hexadecimal data stream in the buffer, since the AS/400 system does not change or validate the values that are passed. This value cannot be specified with SPOOL(*NO).

*AFPDS  An advanced function print data stream (AFPDS) is created. Some systems refer to this data stream as MODCA-P.

*AFPDSLINE  Mixed data (line data and AFPDS data) is created. This value can be specified when using any printer supported by PSF. The printer must be configured with AFP(*YES).

*LINES  Line data is created. This value can be specified when using any printer supported by PSF. The printer must be configured with AFP(*YES).

Page size (PAGESIZE)

Specifies the length and width of the printer forms used by this device file. The length is specified in lines per page or by the units specified for the UOM parameter. The width is specified in print positions (characters) per line or by the units specified for the UOM parameter.

The page size must be specified with reference to the way the data is printed on the page. For example, if using 8.5 inch wide by 11.0 inch long forms and printing at 6 lines per inch with a 10-pitch font, specify PAGESIZE(66 85) PAGRTT(0). However, to rotate the page, specify the page size for an 11.0 inch wide by 8.5 inch long page and enter PAGESIZE(51 110) PAGRTT(90).

Note: Specify PAGRTT(*AUTO) or PAGRTT(*DEVD) and PRTQLTY(*DRAFT) on this command to enable automatic reduction or rotation if the data does not fit on the paper.

Specify PAGRTT(*COR) on this command to enable automatic reduction whether or not the data fits on the paper.

Element 1: Page length

0.001-255.0  Specify the page length used by this device file. Although a value ranging from .001 through 255 is allowed, the value specified should not exceed the actual length of the forms used.

Element 2: Page width
Specifies the page width used by this device file. The value specified should not exceed the actual width of the page used. Valid values for the 3203, 4245, 5211, 5256, 5262, and 3287 printers range from 1 through 132.

Element 3: Measurement method

**ROWCOL**
Page-length and page-width are measured as numbers of rows and columns.

**UOM**
Page-length and page-width are measured in the units specified on the UOM parameter.

---

**Lines per inch (LPI)**

Specifies the line spacing setting on the printer, in lines per inch, used by this device file.

The line spacing on the 5256 printer must be set manually. When the lines per inch (LPI) value on this parameter changes (from the value on the previous printer file), an inquiry message is sent to the message queue associated with the printer that requests a change to the LPI value.

The line spacing on the 4214, 4224, 4230, 4234, 4245, and 5262 Printers is set by a print command. These also allow setting the lines per inch spacing on the control panel of the printer. The lines per inch value must not be set at the printer. If the LPI value is overridden at the control panel, the system overrides the value set with the LPI value of the next printer file received.

3  The line spacing on the printer is 3 lines per inch. This value is valid only for double-byte character set (DBCS) printers.

4  The line spacing on the printer is 4 lines per inch.

6  The line spacing on the printer is 6 lines per inch. This is the default value for this parameter on the CRTPRTF command.

7.5 The line spacing on the printer is 7.5 lines per inch. This value is valid only for double-byte character set (DBCS) printers.

8  The line spacing on the printer is 8 lines per inch.

**Note:** When printing double-byte character set (DBCS) data for a file specified with LPI(8), use double spacing. Otherwise, the DBCS data does not print correctly. Alphanumeric data, however, prints correctly in single spacing when LPI(8) is specified.

9  The line spacing on the printer is 9 lines per inch.

12 The line spacing on the printer is 12 lines per inch.

**Double-byte character set considerations:** When printing a file that has double-byte character set (DBCS) data, specify a DBCS printer (5553, 5583).

---

**Characters per inch (CPI)**

Specifies the printer character density (in characters per inch) used by this device file.
For the printers that support fonts, the value specified in the font special value implies the CPI. If FONT('CPI) is specified, the font used is based on the CPI value. The following diagram describes the default font ID for each CPI value:

<table>
<thead>
<tr>
<th>CPI</th>
<th>FONT ID DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>245</td>
</tr>
<tr>
<td>10</td>
<td>011</td>
</tr>
<tr>
<td>12</td>
<td>087</td>
</tr>
<tr>
<td>13.3</td>
<td>204</td>
</tr>
<tr>
<td>15</td>
<td>222</td>
</tr>
<tr>
<td>16.7</td>
<td>400</td>
</tr>
<tr>
<td>18</td>
<td>252</td>
</tr>
<tr>
<td>20</td>
<td>281</td>
</tr>
</tbody>
</table>

5  Character density is 5 characters per inch.
10 Character density is 10 characters per inch. (This is the default value for this parameter on the CRTPRTF command.)
12 Character density is 12 characters per inch.
13.3 Character density is 13.3 characters per inch. This value is valid only for double-byte character set (DBCS) printers.
15 Character density is 15 characters per inch.
16.7 Character density is 16.7 characters per inch.
18 Character density is 18 characters per inch. This value is valid only for double-byte character set (DBCS) printers.
20 Character density is 20 characters per inch. This value is valid only for double-byte character set (DBCS) printers.

Double-byte character set considerations: When printing a file that has double-byte character set (DBCS) data, specify a DBCS printer (5553, 5583).

Front margin (FRONTMGN)

Specifies the offset, down and across, of the origin from the edge on the front side of the paper. The offsets are in the units of measure specified on the UOM parameter. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

Single values

*DEVD

The no-print border from the printer is used to place the text on the page when printing to a printer configured with AFP(*YES). A margin of 0 is used for IPDS printers without a no-print border, or which are configured with AFP(*NO).

Element 1: Offset down

0.0-57.79

Specify the offset of the origin from the top of the page.
Element 2: Offset across

0.0-57.79

Specify the offset of the origin from the left side of the page.

---

**Back margin (BACKMGN)**

Specifies the offset, down and across, of the origin from the edge on the back side of the paper. The offsets are in the units of measure specified on the UOM parameter. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

**Single values**

*FRONTMGN
   
The offsets specified on the FRONTMGN parameter are used.

*DEVD
   
The no-print border from the printer is used to place the text on the page when printing to a printer configured with AFP(*YES). A margin of 0 is used for IPDS printers without a no-print border, or which are configured with AFP(*NO).

---

Element 1: Offset down

0.0-57.79

Specify the offset of the origin from the top of the page.

---

Element 2: Offset across

offset-across

Specify the offset of the origin from the left side of the page.

---

**Overflow line number (OVRFLW)**

Specifies the line number on the page at which overflow to a new page occurs. Generally, after the specified line is printed, the printer overflows to the next page before printing continues. Overflow is signaled when the specified line number is made the current line, whether printing has occurred on that line or not. The value specified must not exceed the forms length specified in the Page size (PAGESIZE) parameter for the file. Margins specified for the printer file are ignored when determining overflow. More information is in the Basic Printing information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter. This parameter overrides the overflow value specified in the printer file, in the program, or in other called OVRPRTF commands.

1-255 Specify the line number on the current page at which overflow to a new page begins, whether or not printing has occurred on that line. The value specified must not be greater than the page length (PAGESIZE). Margins specified for the printer file are ignored when determining overflow.
Fold records (FOLD)

Specifies whether all positions in a record are printed when the record length exceeds the form width. If so, any portion of the record that cannot be printed on the first line is continued (folded) on the next line or lines until the entire record is printed.

The FOLD parameter is ignored under the following conditions:

- When DEVTYPE(*SCS) is not specified.
- When printing through the OfficeVision program.
- When in the S/36 processing environment.

*YES  Records whose length exceeds the form width are folded on the following lines.

*NO  Records are not folded; if a record is longer than the form width, only the first part of the record that fits on one line is printed.

Double-byte character set considerations: The system ignores this parameter when printing double-byte character set (DBCS) files. The system assumes that DBCS records fit on a printed line. If the record exceeds the form width, the system continues printing the record on the next line.

This parameter overrides the value specified in the printer file, in the program, or in other called OVRPRTF commands.

Unprintable character action (RPLUNPRT)

Specifies whether unprintable characters are replaced and which substitution character (if any) is used. An unprintable character is a character that the printer is unable to print.

Note: If *IPDS is specified in the Printer device type (DEVTYPE) parameter, a hyphen (-) is printed for the unprintable characters. The substitution character is ignored for the 3287 printer.

Single values

*NO  Unprintable characters are not replaced. When an unprintable character is detected, a message is sent to the program.

Element 1: Replace character

*YES  Unprintable characters are replaced. The program is not notified when unprintable characters are detected.

Element 2: Replacement character

A blank is used as the substitution character when an unprintable character is detected and *YES is specified.

X'40'-X'7E'

Specify the substitution character that is used each time an unprintable character is detected if *YES is also specified in this parameter. Any printable EBCDIC character can be specified.

Double-byte character set considerations: The system ignores the chosen replacement character when you specify *YES. Instead, the system replaces unprintable characters as follows:

1. If *YES is also specified in the DBCS extension characters (IGCEXNCHR) parameter, the system replaces unprintable characters with double-byte character set (DBCS) underscores.
2. If *NO is specified in the IGCEXNCHR parameter, the system replaces all extension characters with the undefined character.
Align page (ALIGN)

Specifies whether the forms must be aligned in the printer before printing is started. If *YES is specified and *NO is specified in the Spool the data (SPOOL) parameter, and forms alignment is required, the system sends a message to the message queue specified for the printer, and waits for a reply to the message. If *YES is specified on the SPOOL parameter, and *FILE is specified on the Align page (ALIGN) parameter, of the Start Printer Writer (STRPRTWTR) command, this parameter is used to determine whether an alignment message should be sent by the system.

This parameter is ignored when cut sheets are used (spooled and direct output). Page alignment can be done only for text-only files. Page alignment cannot be done for print jobs containing graphics or bar codes.

*NO    No forms alignment is required.
*YES   The forms are aligned before the output is printed.

Source drawer (DRAWER)

Specifies the source drawer used when single-cut sheets are fed into the printer. *AUTOCUT must be specified on the Form feed (FORMFEED) parameter.

*E1    The envelopes are fed from the envelope drawer on the sheet-feed paper handler.
*FORMDF  The paper is fed from the source drawer specified in the form definition. If a form definition is not specified, then source drawer 1 is used.
1-255  Specify the drawer from which the paper is fed.

Output bin (OUTBIN)

Specifies the destination of the output on printers capable of multiple output bins.

*DEVD  The destination of the output is the device default output bin.
1-65535 Specify the output bin for the destination of the output.

Font specifications (FONT)

Specifies the font identifier and point size used with this printer device file. If a font identifier and point size is not specified, the system automatically sets them.

Single values

*CPI    The identifier of the font with the specified pitch (characters per inch (CPI)) is used.
*DEVD
  The font identifier and point size specified in the device description are used.

Element 1: Identifier

identifier
  Specify the numeric font identifier being used with this printer device file.

Element 2: Point size

*NONE
  No point size is specified; the system sets one based on the type of printer being used.

0.1-999.9
  Specify a point size.

---

Form feed (FORMFEED)

Specifies the form feed attachment used by this printer device file.

*DEVD
  The forms are fed into the printer in the manner specified in the device description.

*CONT
  Continuous forms are used by the printer. The tractor-feed attachment must be put on the printer if this value is specified.

*CONT2
  Continuous forms are used by the printer. The form is fed from the secondary tractor feed attachment. The secondary tractor feed attachment must be on the printer device.

*CUT
  Single-cut sheets are used by the printer. Each sheet must be loaded manually. For cut sheets, the forms alignment message is not issued.

*AUTOCUT
  Single-cut sheets are semiautomatically fed into the printer. The sheet-feed attachment must be put on the printer if this value is specified. For cut sheets, the forms alignment message is not issued.

---

Print quality (PRTQLTY)

Specifies, for the 3812 SCS, 3816 SCS, 4214, 4224, 4230, 4234, and 5219 printers, the quality of print produced.

For the 5219 Printer, different print qualities are produced by varying the speed at which the print ribbon advances. Quality mode (*STD or *NLQ) results in normal print ribbon advancement. In draft mode (*DRAFT), the ribbon advances at a rate of one-third the distance it advances in quality mode. In other words, the 5219 Printer conserves printer ribbon when in draft mode by not advancing it as fast per character printed. The 5219 Printer has a conserve ribbon switch that overrides the value of *DRAFT specified by this parameter.

For the 3812 SCS and 3816 SCS Printers, the automatic hardware selection of computer output reduction printing selected through soft switches on the printers occurs only when *DRAFT is specified for PRTQLTY and PAGRTT is *DEVD. If PAGRTT(*COR) is specified, the PRTQLTY parameter does not affect the printed output.
For the 4224, 4230, and 4234 Printers, standard print quality is produced by varying the density of the
dot matrix pattern used to create printable characters. Standard mode (*STD) is the normal mode. Quality
mode (*NLQ) requires multiple passes by the printer to produce a line of data. Draft mode (*DRAFT)
results in high-speed printing.

For the 4214 printer, only draft (*DRAFT), quality (*NLQ), and device default (*DEVD) modes are
supported. Other values are set to quality (*NLQ) mode.

**NOTES:**
- For the 4214 Printer, quality mode (*STD or *NLQ) is only supported for 10 and 12 characters per inch.
  If PRTQLTY(*STD or *NLQ) and 5, 15, or 16.7 characters per inch is specified, the data is printed in
draft mode.
- For the 4234 Printer, only a limited character set (62 characters) is supported when PRTQLTY(*DRAFT)
is specified. A description of the character set supported with draft print quality is in the 4234 Printer
  Operator’s Guide.
- For the 4224 and 4230 printers, the fonts supported are not available for all three print qualities. The
OCR-A and OCR-B fonts are supported only with PRTQLTY(*NLQ). The Courier and Essay fonts are
available only with PRTQLTY(*NLQ) and PRTQLTY(*STD). The Gothic font is available only with
PRTQLTY(*DRAFT) or PRTQLTY(*FASTDRAFT). If there is a mismatch between the print quality and
the font selected, the font is changed to match the print quality.
- Specify PAGRTT(*DEVD) and PRTQLTY(*DRAFT) on this command to enable automatic rotation if the
data does not fit on the paper.

*STD The output is printed with standard quality.

*DRAFT The output is printed with draft quality.

*DEVD The print quality is set on the printer by the user. It is not set in the data stream.

*NLQ The output is printed with near letter quality.

*FASTDRAFT The output is printed at a higher speed and with lower quality than it would be if you specified
*DRAFT. This value is only supported by the 4230 printer.

**Control character (CTLCHAR)**

Specifies whether the printer device file supports input with print control characters. Any control
characters found that are not valid are ignored, and single spacing is assumed.

*NONE No print control characters are passed in the data printed.

*FCFC The first character of every record contains an ANSI forms-control character. If *FCFC is specified,
the record length must include one position for the first-character forms-control code. This value
is not valid for externally described printer files.

*MACHINE The first character of every record contains a machine code control character. If *MACHINE is
specified, the record length must include one extra position for the first character forms control
code. This value is not valid for externally described printer files.

If TBLREFCHR(*YES) is also specified, then the record length must include two extra positions
for the control character and the table reference character.
Channel values (CHLVAL)

Specifies a list of channel numbers with their assigned line numbers. Use this parameter only if *FCFC is specified in the Control character (CTLCHAR) parameter.

Note: If one or more channel-number/line-number combinations are changed, all other combinations must be re-entered.

Single values

*NORMAL
The default values for skipping to channel identifiers are used.

Element 1: Channel

1-12 Specify an American National Standard channel number to be associated with a corresponding ‘skip to’ line number. Values for this parameter correspond to channels 1 through 12. The CHLVAL parameter associates the channel number with a page line number. For example, if you specify CHLVAL(2 20), channel identifier 2 is allocated with line number 20; therefore, if you place the forms-control 2 in the first position of a record, the printer skips to line 20 before printing the line.

Note: If the printer stops and the next record processed has a channel value forms-control number that is the same value as the line number the printer is on, the printer advances to that value (line number) on the next page. However, if the printer is positioned at the top of the page (line number one) and the channel value forms-control value is associated with line number one, the printer does not advance to a new page.

If no line number is specified for a channel identifier, and that channel identifier is encountered in the data, a default of ‘space one line’ before printing is used. Each channel number can be specified only once.

Element 2: Line number for channel

Element 1: Line

1-255 Specify the line number assigned for the channel number in the same list. Valid line numbers range from 1 through 255. If no line number is assigned to a channel number, and that channel number is encountered in the data, a default of ‘space one line’ before printing is used. Each line number should be specified only once.

Fidelity (FIDELITY)

Specifies whether printing continues when print errors are found for printers configured with AFP(*YES).

*CONTENT
Printing continues when errors are found.

*ABSOLUTE
Printing stops when errors are found.
Character identifier (CHRID)

Specifies the character identifier (graphic character set and code page) for the file. This parameter allows you to print text that is in different character identifier (graphic character set and code page) coding. The value specified on this parameter is used to command the printer device to interpret the hexadecimal byte string by printing the same characters that were intended when the text was created.

Single values

*DEVD
The default CHRID value that the device is designed to handle is used. Character selection is normal because the file has the same character identifier as the device default.

*SYSVAL
The CHRID value specified for the system on which the application runs is used.

*JOBCCSID
The character identifier for the printer file is taken from the coded character set identifier (CCSID) of the job.

Note: This value is not allowed if the file was created on a system at an earlier release level than V2R3M0.

Element 1: Graphic character set

integer
Specify the graphic character set value that matches the printer. Valid values range from 1 through 32767.

Element 2: Code page

integer
Specify the code page value that matches the printer. Valid values range from 1 through 32767.

Decimal format (DECFMT)

Specifies which decimal format value is used when editing numeric fields with the EDTCDE DDS keyword. The decimal format value determines the use of commas and periods for the decimal position and three digit positional separators on edited fields.

*FILE Use the decimal format value stored with the file when the file was created.

*JOB Use the decimal format value from the DECFMT job attribute when the file is opened.

Font character set (FNTCHRSET)

Specifies a downloaded font consisting of a character set and code page. For an outline font, a point size is required. For a raster font, the point size is ignored. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

Single values
*FONT
The value specified on the FONT parameter is used.

Element 1: Character set

Qualifier 1: Character set
name Specify the name of the font character set.

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 font character set. 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 character set is located.

Element 2: Code page

Qualifier 1: Code page
name Specify the name of the code page.

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 code page name. 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 code page name is located.

Element 3: Point size

*NONE The point size is supplied by the system and is determined by the specified font character set.
0.1-999.9 Specify a point size.
**Coded font (CDEFNT)**

Specifies the coded font that the system uses for single-byte character set (SBCS) printing. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

**Single values**

*FNTCHRSET

The font specified on the FNTCHRSET parameter is used.

**Element 1: Coded font**

**Qualifier 1: Coded font**

name Specify the coded font name to use.

**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 coded font name. 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 coded font name is located.

**Element 2: Point size**

*NONE The point size is supplied by the system and is determined by the specified font character set.

0.1-999.9 Specify a point size.

**Page definition (PAGDFN)**

Specifies the page definition to be used to format line data.

You can specify a page definition with *LINE, *AFPDSLNE, or *USERASCII data. PSF/400 will convert the line data and page definition to IPDS.

When you specify a page definition on the printer file, some printer file parameters will be ignored when the spooled file is printed by PSF/400. The following print file parameters will be ignored:

- CDEFNT
- CHRID
- CPI
- FNTCHRSET
- FOLD
- FONT
• LPI
• MULTIUP
• PAGESIZE
• PAGRTT
• REDUCE

Single values

*NONE
No page definition is specified.

Because PSF/400 requires a page definition when *LINE or *AFPDSLINE is specified, an inline page definition is built from the print file parameters and passed to PSF/400 when *NONE is specified.

Qualifier 1: Page definition

name Specify the name of the page definition that must exist in the library specified. Valid values range from 1 to 8 characters. Device type *AFPDSLINE, *LINE, or *USERASCII must be specified when using a page definition.

Qualifier 2: Library

*Ltbl 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.

Form definition (FORMDF)

Specifies the form definition to use when printing the file. A form definition is a resource object that defines the characteristics of the form, including overlays, position of page data on the form, and number of copies of pages and modifications to pages. The form definition is located inline with the file being printed, or in a library.

When you specify a form definition (*DEVD or form definition name) on the printer file, some printer file parameters will be ignored when the spooled file is printed by PSF/400. The following print file parameters will be ignored:
• DUPLEX (If *FORMDF specified)
• DRAWER (If *FORMDF specified)
• PAGRTT
• PRTQLTY
• FORMFEED
• FRONTMGN
• BACKMGN
• MULTIUP
• REDUCE
• CORNERSTPL
• EDGESTITCH
• SADLISTITCH

Single values

*NONE

No form definition is used.

Because PSF/400 requires a form definition, an inline form definition is built from the print file parameters and passed to PSF/400 when *NONE is specified.

*DEVD

The name of the form definition is specified in the printer device description.

Qualifier 1: Form definition

name Specify the name of the form definition that must exist in the library specified. Valid values range from 1 to 8 characters.

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.

AFP Characters (AFPCHARS)

Specifies one or more AFP characters (coded fonts) to be used with line data and a page definition.

Single values

*NONE

No AFP characters (coded fonts) specified.

Other values (up to 4 repetitions)

name Specify up to four 4-byte names of coded fonts to be specified with the line data and a page definition. The 4-byte names are concatenated to X0 to identify up to four coded fonts which are to be used when TBLREFCHR is being used within the data.

Table Reference Characters (TBLREFCHR)

Specifies whether table reference characters are present in the line data.

*NO No table reference character is present in line data.

*YES Table reference characters are present in line data.

If forms control characters are used with the data, the table reference character follows the forms control character but precedes the data bytes. If forms control characters are not used, the table reference character is the first byte of the data record. As with forms control character, if table reference characters are used, every data record must contain a TRC byte.
**Degree of page rotation (PAGRTT)**

Specifies the degree of rotation of the text on the page with respect to the way the form is loaded into the printer.

*AUTO*  
Indicates that automatic rotation of output is done to fit the printed data on the form. If rotation does not accomplish this, computer output reduction is performed automatically (regardless of the print quality being used). This parameter is valid only for printers supporting rotation.

*DEVD*  
The operating system sends a device default rotation value to the printer. Page rotation is dependent on your printer’s specifications. See your printer or printer emulation documentation to determine how page rotation is affected.

*COR*  
Computer output reduction is done. Computer output reduction allows printed output intended for a 13.2-inch wide by 11.0-inch long form to be printed on an 8.5-inch wide by 11.0-inch long form.

0  
No rotation is done. Printing starts at the edge loaded into the printer first, and is parallel to that edge.

90  
Rotation of the text is done 90 degrees clockwise from the 0 degree writing position.

180  
Rotation of the text is done 180 degrees clockwise from the 0 degree writing position.

270  
Rotation of the text is done 270 degrees clockwise from the 0 degree writing position.

**Pages per side (MULTIUP)**

Specifies, for spooled output only, whether multiple pages of output are printed on 1 physical page.

1  
One page of output is printed on one physical sheet of paper.

2  
Two pages of output are printed on 1 physical sheet of paper.

3  
Three pages of output are printed on 1 physical sheet of paper.

4  
Four pages of output are printed on 1 physical sheet of paper.

**Reduce output (REDUCE)**

Specifies whether to reduce the output when doing multiple up printing.

For examples and more details see the Basic Printing information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

*TEXT*  
The text output is reduced when doing multiple up printing.

*NONE*  
The output is not reduced when doing multiple up printing.
Print text (PRTTXT)
Specifies the text that is printed at the bottom of each page of printed output and on separator pages.

*JOB The value from the current job is used.
*BLANK No text is printed.

character-value
Specify no more that 30 characters of text, enclosed in apostrophes.

Hardware justification (JUSTIFY)
Specifies the printing positions of the characters on a page so that the right-hand margin of printing is regular.

0 No justification occurs. (This is the default value for this parameter on the CRTPRTF command.)
50 Spaces are added to the blanks in the text so that the right margin is more closely aligned, but not flush.
100 The text is expanded by spaces (added where the blanks already exist) until the right margin is flush.

Print on both sides (DUPLEX)
Specifies whether output is printed on one side or two sides of the paper.

*NO The output is printed on one side of the paper.
*YES The output is printed on both sides of the paper, with the top of each printed page at the same end of the sheet of paper.
*TUMBLE The output is printed on both sides of the paper, with the top of one printed page at the opposite end from the top of the other printed page. This is usually used for output that will be bound at the top.
*FORMDF The output is printed on both sides of the paper if the duplex value is specified in the form definition. If a form definition is not specified, then the output is printed on one side of the paper.

Unit of measure (UOM)
Specifies the unit of measurement to be used.

*INCH The inch is used as the unit of measurement.
*CM The centimeter is used as the unit of measurement.
Front side overlay (FRONTOVL)

Specifies the qualified name of the object that contains both the overlay that is printed on the front side of the page and the offset, down and across, from the point of origin used when the overlay is printed.

Single values

*NONE
   No overlay is used.

Element 1: Overlay

Qualifier 1: Overlay

name Specify the name of the overlay.

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 overlay. 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 overlay is located.

Element 2: Offset down

0 No offset down from the point of origin is used.

0.0-57.79 Specify the offset down from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

Element 3: Offset across

0 No offset across from the point of origin is used.

0.0-57.79 Specify the offset across from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

Back side overlay (BACKOVL)

Specifies the object name and library name containing both the overlay that is printed on the BACK side of the page and the offset, down and across, from the point of origin used when the overlay is printed.

The constant back function allows you to print overlays on blank pages without adding blank pages to the print application. Specifying the constant back function would cause, for each page generated by the application program, a blank page to be generated onto which the specified back overlay could be
The generated blank pages are called constant forms because no variable data from the user’s program is printed on the pages. The constant back function is only supported for duplex printing. It is ignored when DUPLEX(*NO) is specified on the printer file.

Note that the offset down and offset across values are ignored when *CONSTANT is specified for constant back. An offset of 0.0 is assumed for these values.

**Single values**

*FRONTOVL*

The values that are specified on the FRONTOVL parameter are used.

*NONE*

No overlay is used.

**Element 1: Overlay**

**Qualifier 1: Overlay**

*name* Specify the name of the overlay.

**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 overlay. 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 overlay is located.

**Element 2: Offset down**

0 No offset down from the point of origin is used.

0.0-57.79 Specify the offset down from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

**Element 3: Offset across**

0 No offset across from the point of origin is used.

0.0-57.79 Specify the offset across from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

**Element 4: Constant back**

*NOCONSTANT* No constant back is specified.

*CONSTANT* Constant back is specified.
**Convert line data (CVTLINDTA)**

Specifies whether line data and a page definition should be converted to AFPDS before the data is spooled.

*NO  No AFPDS conversion is done.

*YES  Specifies that AFPDS conversion is to be done on the line data and page definition before the data is spooled.

**IPDS pass through (IPDSPASTHR)**

Specifies whether IPDS (intelligent printer data stream) pass-through is done for the spooled file.

*DEVD  
The value specified for IPDSPASTHR in the PSF configuration object specified for a printer device description is used. If no PSF configuration object is specified for the device, a value of *NO is used.

*NO  No IPDS pass-through is done.

*YES  Specifies that IPDS pass-through is to be done if the spooled file is eligible for IPDS pass-through.

**Note:** Not all SCS or IPDS spooled files are eligible for IPDS pass-through. They may contain special functions that require transform to AFPDS for correct printing. Specifying IPDS pass-through on the printer file allows only those spooled files eligible for IPDS pass-through to bypass the extra transforms. Those spooled files not eligible for IPDS pass-through will still undergo the transforms to AFPDS and back to IPDS.

IPDS pass-through will not be valid for all PSF/400 supported printers. Any printer (or attachment) that does not support resident fonts can not support IPDS pass-through. This is because the resident font references in the data stream must be mapped to host fonts which are downloaded to the printer. All IBM IPDS printers, except for the following, can be supported with IPDS pass-through: 3820, 3825, 3827, 3828, 3829, 3831, 3835, 3900-001 and any printer attached using Print Services Facility for OS/2’s Distributed Print Function.

For V3R7, V4R1 and V4R2, IPDSPASTHR can be specified with the USRDFNDTA parameter in a printer file. You may continue using this support with existing printer files and PSF configuration objects by specifying IPDSPASTHR(*DEVD) in the printer file. If you specify a value of anything other than *DEVD for the IPDSPASTHR parameter, any IPDS pass-through value in the USRDFNDTA parameter is ignored.

**User resource library list (USRRSCLIBL)**

Specifies the list of user resource libraries to be used for searching for AFP resources for a spooled file. If the AFP resource is not found in the user resource libraries, then the library list specified in the DEVRSCLIBL parameter of the PSF configuration object is searched. If no PSF configuration object is specified for the device, then libraries QFNTCPL, QFNT01-QFNT19, and QFNT61-69 are searched.

**Single values**
*DEVD
The value specified for USRRSCLIBL in the PSF configuration object specified for a printer device description is used. If no PSF configuration object is specified for the device, a value of *JOBLIBL is used.

*NONE
No user libraries are specified.

*JOBLIB
Specifies that the library list of the job that created the spool file is used in searching for AFP resources. This library list is saved with the spool file when it is created.

*CURLIB
Specifies that the current library of the job that created the spool file is used for searching for AFP resources. If no library is specified as the current library for the job, then library QGPL is used.

Other values (up to 4 repetitions)

character-value
Specify the name of a library that will be used to search for AFP resources. Up to four library names may be specified.

For V3R7, V4R1 and V4R2, USRRSCLIBL can be specified with the USRDFNDTA parameter in a printer file. PSF/400 uses that value if USRRSCLIBL(*PRTF) is specified in a PSF configuration object which is specified in the printer device description. You may continue using this support with existing printer files and PSF configuration objects by specifying USRRSCLIBL(*DEVD) in the printer file. If you specify a value of anything other than *DEVD for the USRRSCLIBL parameter, any user resource library value in the USRDFNDTA parameter is ignored.

---

Corner staple (CORNERSTPL)

Specifies the reference corner to be used for a corner staple. A staple is driven into the media at the reference corner. Refer to your printer’s documentation for information as to which reference corners are supported. Page rotation does not affect the placement of a corner staple.

*NONE
A corner staple is not specified.

*DEVD
The reference corner is the default reference corner used by the device.

*BOTRIGHT
The reference corner is the bottom right corner of the media.

*TOPRIGHT
The reference corner is the top right corner of the media.

*TOPLEFT
The reference corner is the top left corner of the media.

*BOTLEFT
The reference corner is the bottom left corner of the media.
Edge stitch (EDGESTITCH)

Specifies where one or more staples are driven into the media along the finishing operation axis. Refer to your printer’s documentation for information about which elements of this parameter are supported and which values for each element are supported. If specification of a value for an element is not supported by a printer, specify a value of *DEVD for that element. Page rotation does not affect the placement of an edge stitch.

**Single values**

*NONE
   An edge stitch is not specified.

**Element 1: Reference edge**

Specifies the reference edge to be used for an edge stitch. An edge stitch is formed by having one or more staples driven into the media along the finishing operation axis.

*DEVD
   The reference edge is the default reference edge used by the device.

*BOTTOM
   The reference edge is the bottom edge of the media.

*RIGHT
   The reference edge is the right edge of the media.

*TOP
   The reference edge is the top edge of the media.

*LEFT
   The reference edge is the left edge of the media.

**Element 2: Reference edge offset**

Specifies the offset of the edge stitch from the reference edge toward the center of the media.

*DEVD
   The reference edge offset is the default reference edge offset used by the device.

0.0-57.79
   Specifies the offset of the edge stitch from the reference edge. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This value is converted to millimeters for the printer. Fractional millimeters are not supported and are discarded when conversion to millimeters is performed.

**Element 3: Number of staples**

Specifies the number of staples that are to be applied along the finishing operation axis.

*DEVD
   The number of staples depends on the value of the Staple Offsets element of this parameter. If *DEVD is also specified or defaulted for the Staple Offsets element value, then the number of staples is the default number of staples used by the device. If one or more offsets are specified for Staple Offsets, the number of staples is the same as the number of staple offsets specified.

1-122
   Specify the number of staples to be used for the edge stitch. If the number of staples is specified, then *DEVD must be specified for staple offsets. The device default for the spacing of each staple will be used.

**Element 4: Staple offsets**

Override with Printer File (OVPRPTF) 121
Specifies the offset of the staples along the finishing operation axis. The offset is measured from the point where the finishing operation axis intersects either the bottom edge or the left edge of the media, toward the center of the media. Each consecutive value is used to position a single finishing operation centered on the specified point on the finishing operation axis.

*DEVD
The staple offsets are the default staple positions used by the device. If a value was specified for the Number of Staples element, the staple position of each staple will be calculated automatically by the printer.

0.0-57.79
Specify the staple offset for each staple in the edge stitch. Up to 122 staple offsets may be specified. If one or more staple offset values are specified, then *DEVD must be specified for the number of staples. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This value is converted to millimeters for the printer. Fractional millimeters are not supported and are discarded when when conversion to millimeters is performed.

Saddle stitch (SADLSTITCH)

Specifies where one or more staples are driven into the media along the finishing operation axis, which is positioned at the center of the media parallel to the reference edge. Page rotation does not affect the placement of a saddle stitch.

Single values

*NONE
A saddle stitch is not specified.

Element 1: Reference edge

Specifies the reference edge to be used for a saddle stitch. A saddle stitch is formed by having one or more staples driven into the media along the finishing operation axis, which is positioned at the center of the media parallel to the reference edge.

*DEVD
The reference edge is the default reference edge used by the device.

*TOP The reference edge is the top edge of the media.

*LEFT The reference edge is the left edge of the media.

Element 2: Number of staples

Specifies the number of staples that are to be applied along the finishing operation axis.

*DEVD
The number of staples depends on the value of the Staple Offsets element of this parameter. If *DEVD is also specified or defaulted for the Staple Offsets element value, then the number of staples is the default number of staples used by the device. If one or more offsets are specified for Staple Offsets, the number of staples is the same as the number of staple offsets specified.

1-122 Specify the number of staples to be used for the saddle stitch. If you specify the number of staples, then *DEVD must be specified for staple offsets. The device default for the spacing of each staple will be used.

Element 3: Staple offsets

122 IBM Systems - iSeries: i5/OS Commands Starting with MRGTCMPT (Merge TCP/IP Host Table)
Specifies the offset of the staples along the finishing operation axis. The offset is measured from the point where the finishing operation axis intersects either the bottom edge or the left edge of the media, toward the center of the media. Each consecutive value is used to position a single finishing operation centered on the specified point on the finishing operation axis.

*DEVD
The staple offsets are the default staple positions used by the device. If a value was specified for the Number of Staples element, the staple position of each staple will be calculated automatically by the printer.

0.0-57.79
Specify the staple offset for each staple in the saddle stitch. Up to 122 staple offsets may be specified. If one or more staple offset values are specified, then *DEVD must be specified for the number of staples. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This value is converted to millimeters for the printer. Fractional millimeters are not supported and are discarded when when conversion to millimeters is performed.

---

**Font resolution for formatting (RNTRSL)**

Specifies the resolution PSF/400 uses when printing to a multiple resolution printer configured to report multiple resolutions, but the spooled file does not specify the font metrics and resolution or the font is not available at the resolution that is contained in the spooled file.

For more information regarding the algorithm used for searching a library list for a font resource, see the Printer Device Programming manual section entitled User and Device Resource Library Lists in the chapter called Working With PSF configuration objects.

*DEVD
The value specified in the FNTRSL parameter of the PSF configuration object for the device is used. If no PSF configuration object is specified for the device, a value of *SEARCH is used.

*SEARCH
Specifies to search the library list for the first occurrence of a host font with a name match. The resolution of that font is used to print the spool file. Message PQT3546 is sent to specify the resolution of the font that was selected.

240 The font resolution is 240 pels per inch.

300 The font resolution is 300 pels per inch.

---

**Defer write (DFRWRT)**

Specifies whether output is held in the system buffer before being sent to the printer.

*YES The system controls the amount of output that is held in the buffer before it is sent to the printer.

*NO If *NO is specified on this parameter and if *NO is specified on the Spool the data (SPOOL) parameter, output is not held in the buffer. Instead, output is sent immediately to the printer once the program has performed a write operation.

If *NO is specified on this parameter and if *YES is specified on the SPOOL parameter and if *IMMED is specified on the Spooled output schedule (SCHEDULE) parameter, output is held in the buffer until a page of output is available or until the system buffer is full.
**Spool the data (SPOOL)**

Specifies whether the output data for the printer device file is spooled.

*YES  The data is spooled for processing by a diskette writer or printer writer.

*NO   The data is not spooled. It is sent directly to the device to print as the output becomes available.

**Output queue (OUTQ)**

Specifies the output queue used for spooled files that specify OUTQ(*JOB). This parameter applies only to printer files that have *JOB specified for the OUTQ parameter.

**Single values**

*DEV   The output queue associated with the printer specified on the DEV parameter is used. The output queue has the same name as the printer.

*JOB   The output queue associated with this job is used for the spooled output.

**Qualifier 1: Output queue**

*name   Specify the name of the output queue to which the output data is spooled.

**Qualifier 2: Library**

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

*CURLIB The current library for the job is used to locate the output queue. If no library is specified as the current library for the job, QGPL is used.

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

**Form type (FORMTYPE)**

Specifies the type of forms used in the printer for printed output that is produced using this device file. If a form type other than *STD is specified, the system sends a message that identifies the form type to the system operator when the output is produced, and requests that the specified type of forms be put in the printer. This parameter overrides the form type value specified in the printer file or in other called OVRPRFT commands.

*STD   The standard printer form for your computer system is used.

*character-value  Specify a form type identifier, having 10 characters or less, for the printer forms used.

**Copies (COPIES)**

Specifies, for spooled output only, the number of copies of the output being printed.
Page range to print (PAGERANGE)

Specifies, for spooled output files only, the starting and ending pages to print.

Element 1: Starting page

*ENDPAGE
  Use the end page value as the starting page.

integer
  Specify the starting page to print.

Element 2: Ending page

*END
  Printing continues until the end of file.

integer
  Specify the ending page to print.

Max spooled output records (MAXRCDS)

Specifies, for spooled output only, the maximum number of records that can be in the spooled file for jobs using the printer file. This parameter overrides the value specified in the printer file or in other called OVRPRTF commands.

*NOMAX
  There is no maximum on the number of records that can be in the spooled file.

1-999999
  Specify the maximum number of records that can be in the spooled output file.

File separators (FILESEP)

Specifies, for spooled output only, the number of separator pages placed at the beginning of each printed file, including the pages between multiple copies of the same output. Each separator page has the following items printed on it: file name, file number, job name, user name, and the job number. This parameter overrides the separator value specified in the printer file or in other called OVRPRTF commands.

0-9
  Specify the number of separator pages used at the start of each printed output file produced by this device file. If 0 is specified, no separator pages are printed for the file. In this case, the printed output for each file (or copy of a file) starts at the top of a new page.

Spooled output schedule (SCHEDULE)

Specifies, for spooled output files only, when the spooled output file is made available to a spooling writer. This parameter overrides the scheduling value specified in the printer file or in other called OVRPRTF commands.
*JOBEND

The spooled output file is available to the spooling writer only after the entire job is completed.

*FILEEND

The spooled output file is available to the spooling writer as soon as the file is closed in the program.

*IMMED

The spooled output file is made available to the writer as soon as the file is opened in the program.

---

**Hold spooled file (HOLD)**

Specifies, for spooled output files only, whether the spooled file is held. The spooled file can be released by using the Release Spooled File (RLSSPLF) command.

**Note:** This parameter overrides the hold value specified in the printer file or in other called OVRPRTF commands.

*NO  The spooled output file is not held on the output queue. The spooled output is available to a spooling writer based on the Spooled output schedule (SCHEDULE) parameter value.

*YES  The spooled output file is held until it is released by the Release Spooled File (RLSSPLF) command.

---

**Save spooled file (SAVE)**

Specifies, for spooled output only, whether the spooled file is saved after the output is produced. This parameter overrides the save value specified in the printer file or in other called OVRPRTF commands.

*NO  The spooled file data is not kept (saved) on the output queue after it is produced.

*YES  The spooled file data is kept on the output queue until the file is deleted. After the file is produced, the number of copies is set to 1, and the status of the file is changed from WTR to SAV.

---

**Output priority (on OUTQ) (OUTPTY)**

Specifies the output priority for spooled output files that are produced by this job. The highest priority is 1 and the lowest priority is 9.

*JOB  The output priority associated with the job that created the spooled file is used.

**output-priority**

Specify a number ranging from 1 (high) through 9 (low) to indicate the output priority.

---

**User data (USRDTA)**

Specifies, for spooled output, user-specified data that identifies the file.
*SOURCE

If the file was created by a System/36 procedure, the name of the procedure is assigned. If the file was created by an application program, the name of the program is assigned.

character-value

Specify up to 10 characters of user-specified text.

---

Spool file owner (SPLFOWN)

Specifies, for spooled output only, who the owner of the spooled file is.

*CURUSRPRF

The spooled file is owned by the current effective user of the current job or thread. See the Basic Printing information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more detailed information on how the SPLFOWN parameter is affected when using any of the following APIs:

- QWTSETP - Set Profile
- qsysuserid() - Set User ID
- qsysseteuid() - Set Effective User ID
- qsysetreuid() - Set Real and Effective User ID

*JOB

The spooled file is owned by the original user profile of the job. If the job has switched to a new user profile, the original user profile is still the owner of the spooled file.

*CURGRPPRF

The spooled file is owned by the current effective group profile of the current job or thread. If there is no current effective group profile, ownership of the spooled file is determined in the same manner as *CURUSRPRF. See the Basic Printing information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more detailed information on how the SPLFOWN parameter is affected when using any of the following APIs:

- QWTSETP - Set Profile
- qsysgetgid() - Set Group ID
- qsyssetegid() - Set Effective Group ID
- qsysetre gid() - Set Real and Effective Group ID

*JOBGRPPRF

The spooled file is owned by the group profile of the original user profile of the job. If the job has switched to a new user profile, the group profile of the original user profile is still the owner of the spooled file. If no group profile exists, ownership of the spooled file is determined the same way as *JOB.

User Defined Option (USRDFNOPT)

Specifies, for spooled output only, one or more user-defined options to be used by user applications or user-specified programs that process spooled files. A maximum of four user-defined options can be specified.

This parameter overrides the user-defined options specified in the printer file or in other called OVRPRTF commands.

Single values
*NONE
   No user-defined options specified.

Other values (up to 4 repetitions)
character-value
   Specify a user-defined option to be used by user applications or user-specified programs that process spooled files. All characters are acceptable.

---

**User Defined Data (USRDFNDTA)**

Specifies, for spooled output only, the user-defined data to be used by user applications or user-specified programs that process spooled files.

This parameter overrides the user-defined data specified in the printer file or in other called OVRPRTF commands.

*NONE
   No user-defined data specified.

character-value
   Specify a user-defined data to be used by user applications or user-specified programs that process spooled files. All characters are acceptable.

---

**User Defined Object (USRDFNOBJ)**

Specifies, for spooled output only, the user-defined object to be used by user applications or user-specified programs that process spooled files.

This parameter overrides the user-defined object name specified in the printer file or in other called OVRPRTF commands.

**Single values**

*NONE
   No user-defined object specified.

**Element 1: Object**

   **Qualifier 1: Object**

   name Specify the name of the user-defined object to be used by user applications or user-specified programs that process spooled files.

   **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.

Element 2: Object type

object-type

The user object type can be one of the following:

*DTAARA Data Area
*DTAQ Data Queue
*FILE File
*PSFCFG PSF Configuration Object
*USRIDX User Index
*USRQ User Queue
*USRSPC User Space

Spool file name (SPLFNAME)

Specifies, for spooled output only, the spooled output file name.

*FILE The name of the printer file is used for the spooled output file name.

name Specify the name of the spooled output file. A maximum of 10 characters can be used.

Expiration date for file (EXPDATE)

Specifies the expiration date for the spooled file. The spooled file will expire at 23:59:59, system local time on the date specified.

*NONE No expiration date is specified.

*DAYS The expiration date is to be calculated using the value specified for the Days until file expires (DAYS) parameter.

date Specify the date after which the spooled file will be eligible for removal from the system by the Delete Expired Spooled Files (DLTEXP SPLF) command. The date must be enclosed in apostrophes if date separator characters are used in the value.
Days until file expires (DAYS)

Specifies the number of days to keep the spooled file.

Note: A value must be specified for this parameter if the Expiration date for file (EXPDATE) parameter has a value of *DAYS. If the EXPDATE parameter has a value other than *DAYS, no value is allowed for this parameter.

1-366 Specify an interval in days after which the spooled file will be eligible for removal from the system by the Delete Expired Spooled Files (DLTEXPSPLF) command. The actual expiration date applied to the spooled file is calculated by adding the number of days specified to the date the printer file is opened.

User specified DBCS data (IGCDTA)

Specifies, for program-described files, whether the file processes double-byte character set (DBCS) data. Specifies, for externally described files, the DBCS attributes of the file.

For program-described files:

*NO The file does not process double-byte character set (DBCS) data.
*YES The file processes double-byte character set (DBCS) data.

For externally-described files:

*NO The only double-byte character set (DBCS) attributes of the file are those defined in the DDS.
*YES DBCS attributes, in addition to those defined in the DDS, include putting the DDS keyword for alternative data type (IGCALTTYP) into effect, and identifying DBCS attributes of fields or messages not identified in the DDS.

DBCS extension characters (IGCEXNCHR)

Specifies whether the system processes double-byte character set (DBCS) extension characters.

*YES The system processes extension characters.
*NO The system does not process extension characters; it prints them as undefined characters.

DBCS character rotation (IGCCHRRTT)

Specifies whether the printer should rotate double-byte character set (DBCS) data 90 degrees counterclockwise when printing. The system prints rotated DBCS characters vertically so that they appear in proper reading sequence. Alphanumeric characters are not rotated.

*NO The system does not rotate DBCS data when printing.
*YES The system rotates DBCS data 90 degrees counterclockwise when printing.
**DBCS characters per inch (IGCCPI)**

Specifies the printer character density of double-byte character set (DBCS) data, in characters per inch (cpi).

* **CPI**
  DBCS density is based on the values specified for the Characters per inch (CPI) parameter. The system prints one double-byte character for every two alphanumeric characters. (*CPI is the default value for this parameter on the CRTPRTF command.)*
  - For CPI(10), DBCS characters print at 5 characters per inch.
  - For CPI(12), DBCS characters print at 6 characters per inch.
  - For CPI(13.3), DBCS characters print at 6.7 characters per inch (same as IGCCPI(*CONDENSED)).
  - For CPI(15), DBCS characters print at 7.5 characters per inch.
  - For CPI(18), DBCS characters print at 9 characters per inch.
  - For CPI(20), DBCS characters print at 10 characters per inch.

* **CONDENSED**
  Condensed printing, where the system prints 20 double-byte characters every three inches, is used.

  5  DBCS density is 5 characters per inch.
  6  DBCS density is 6 characters per inch.
  10 DBCS density is 10 characters per inch.

**DBCS SO/SI spacing (IGCSOSI)**

Specifies how the system prints double-byte characters.

* **YES**
  The system prints double-byte characters as blanks.

* **NO**
  The system does not print double-byte characters. These characters do not occupy a position on printed output.

* **RIGHT**
  The system prints two blanks when printing shift-in characters, but does not print shift-out characters.

**DBCS coded font (IGCCDEFNT)**

Specifies the coded font that the system uses for double-byte character set (DBCS) printing. This parameter is only used when using printers configured with AFP(*YES).

**Single values**

* **SYSVAL**
  The DBCS coded font specified in the system value is used.

**Element 1: DBCS coded font**
Qualifier 1: DBCS coded font

name Specify the DBCS coded font name to use.

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 coded font name. 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 coded font name is located.

Element 2: Point size

*NONE The point size is supplied by the system and is determined by the specified font character set.

0.1-999.9 Specify a point size.

Maximum file wait time (WAITFILE)

Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

*IMMED The program does not wait. Immediate allocation of file resources is required.

*CLS The default wait time specified in the class description is used as the wait time for the file resources to be allocated.

integer Specify the number of seconds that the program waits for the file resources to be allocated. Valid values range from 1 through 32767.

Record format level check (LVLCHK)

Specifies whether the level of the device file is checked when the file is opened by a program. For this check, which is done while the file is opened, the system compares the record format identifiers of each record format used by the program with the corresponding identifiers in the device file. Because the same record format name can exist in more than one file, each record format is given a unique internal system identifier when the format is created. Level checking cannot be done unless the program contains the record format identifiers. This command cannot override level checking from *NO to *YES.

*NO The level identifiers are not checked when the file is opened.
Secure from other overrides (SECURE)

Specifies whether this file is safe from the effects of previously called file override commands.

*NO  This file is not protected from other file overrides. Its values are overridden by the effects of any previously called file override commands.

*YES This file is protected from the effects of any previously called file override commands.

Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.

*ACTGRPDFEN

The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

*CALLLVL

The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

*JOB The scope of the override is the job in which the override occurs.

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

*NO  The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

*YES The same ODP is shared with each program in the job that also specifies *YES when it opens the file.

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

*ACTGRPDFEN

The scope of the open operation is determined by the activation group of the program that called the OVRPRTF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

*JOB The scope of the open operation is the job in which the open operation occurs.
Examples

Example 1: Printing Output

OVRPRT FILE(PRINTOUT) TOFILE(PRINT3) SPOOL(*YES)
COPIES(5) OUTQ(OUTPUT1)

This command overrides the file named PRINTOUT and uses the printer file named PRINT3 to produce the spooled output on the printer. The output from the program is sent to the OUTPUT1 output queue. Five copies of the spooled file are printed on the printer specified on the Start Printer Writer (STRPRTWTR) command.

Example 2: Rotating Double-Byte Characters

OVRPRT FILE(IGCLIB/IGCPRT) IGCDTA(*YES) IGCCHRRTT(*YES)

This command overrides the IGCPRT printer file, which is stored in the IGCLIB library. The override puts the IGCALTTYP DDS keyword into effect to change character output fields to DBCS fields, and rotates the double-byte characters when printing.

Error messages

*ESCAPE Messages

CPF180C
Function &1 not allowed.

CPF7343
Channel number specified more than once on CHLVAL.
Override with Save File (OVRSAVF)

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

The Override with Save File (OVRSAVF) command is used (1) to override or replace a file named in a program, (2) to override certain attributes of a file that are used by a program, or (3) to override the file named in a program and certain attributes of the overriding file.

This command does not apply to save and restore commands.

More information on overriding files is in the Files and file systems topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>File being overridden</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>TOFILE</td>
<td>Save file</td>
<td>Single values: *FILE</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Save file</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>EXTEND</td>
<td>Extend file</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>POSITION</td>
<td>Starting position in file</td>
<td>Single values: *START</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Retrieve order</td>
<td>*RRN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: *RRN—Record number</td>
<td>*UNSIGNED integer</td>
<td></td>
</tr>
<tr>
<td>WAITFILE</td>
<td>Maximum file wait time</td>
<td>Integer, *IMMED, *CLS</td>
<td>Optional</td>
</tr>
<tr>
<td>SECURE</td>
<td>Secure from other overrides</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OVRSCOPE</td>
<td>Override scope</td>
<td>*ACTGRPDEF, *CALLLVL, *JOB</td>
<td>Optional</td>
</tr>
<tr>
<td>SHARE</td>
<td>Share open data path</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OPNSCOPE</td>
<td>Open scope</td>
<td>*ACTGRPDEF, *JOB</td>
<td>Optional</td>
</tr>
</tbody>
</table>

File being overridden (FILE)

Specifies the save file in the using program to which this override command is applied. The specified file must be a save file when *FILE is specified in the Save file (TOFILE) parameter.
Note: The information in a save file has meaning only to Operating System/400 save and restore; redirecting another type of file to a save file or vice versa is not recommended.

This is a required parameter.

name Specify the name of the save file.

---

**Save file (TOFILE)**

Specifies the save file that is used instead of the file specified on the **File being overridden (FILE)** parameter or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified on this command. The parameters specified on this command override the other values specified in the save file or in the program.

**Single values**

*FILE The save file named in the FILE parameter has certain parameters overridden by the values specified in this command.

**Qualifier 1: Save file**

name Specify the name of the save file that is used instead of the overridden file name.

**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 save file. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the save file is located.

---

**Extend file (EXTEND)**

Specifies, for output operations only, whether new records are added to the end of the data currently in the save file. This option is used to start processing after an application or a system failure. When this operation is completed, the file must contain the image of a single save operation made by a save command, or it may not be possible to restore objects from the save file. This parameter overrides the extend value specified in the program. The sequencing information in the file’s records guarantees that after a system failure, a record cannot be skipped or sent twice.

*NO Records are not added to the end of the specified save file, but they replace existing records in the file. If the save file already contains records, an inquiry message is sent that clears the file or cancels the operation. If no value is specified for this parameter by the program or in an override, this is the default action assumed when the file is opened for output.

*YES New records are added to the end of the records contained in the save file.
**Starting position in file (POSITION)**

Specifies the starting position for getting records from the save file. The first record to get is either at the beginning of the file (*START) or at a particular relative record number position in the file (*RRN). This parameter overrides the value specified in the program.

**Single values**

*START

Get the first record in the file first. If no value is specified for this parameter by the program, or in an override, this is the default action assumed when the file is opened for input.

Element 1: Retrieve order

*RRN The relative record number specified for the second element of this parameter is the first record to get.

Element 2: *RRN—Record number

`unsigned-integer`

Specify the record number (its position from the beginning of the file) of the record that you get first.

**Maximum file wait time (WAITFILE)**

Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

*IMMED

The program does not wait. Immediate allocation of file resources is required.

*CLS The default wait time specified in the class description is used as the wait time for the allocation of the file resources.

`integer`

Specify the number of seconds that the program waits for the file resources to be allocated. Valid values range from 1 through 32767.

**Secure from other overrides (SECURE)**

Specifies whether this file is protected from the effects of file override commands that were previously called.

*NO This file is not protected from other file overrides; its value is overridden by the effects of any file override commands that were previously called.

*YES This file is protected from the effects of any file override commands that were previously called.
Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.

*ACTGRPDFN
The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

*CALLLVL
The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

*JOB
The scope of the override is the job in which the override occurs.

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

*NO
The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

*YES
The same ODP is shared with each program in the job that also specifies *YES when it opens the file.

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

*ACTGRPDFN
The scope of the open operation is determined by the activation group of the program that called the OVRSAVF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

*JOB
The scope of the open operation is the job in which the open operation occurs.

Examples

OVRSAVF  FILE(ONLINE)  POSITION(*RRN 100)  SECURE(*YES)

This command overrides the file named ONLINE so that the first record gotten after the file is opened for input is relative record number 100. The file is also safe from overrides (in previous program calls).

Error messages

*ESCAPE Messages
CPF180C
   Function &1 not allowed.

CPF1892
   Function &1 not allowed.
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPPHT (Merge TCP/IP Host Table)
Override with Tape File (OVRTAPF)

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

The Override with Tape File (OVRTAPF) command is used to (1) override (replace) a file named in a program, (2) override certain attributes of a file that are used by a program, or (3) override the file named in a program and override certain attributes of the file processed. Parameters overridden by this command are specified in the file description, in the program, or in other called file override commands.

If a file named in the program is overridden, the name of that file is specified in the FILE parameter and the name of the overriding file is specified in the TOFILE parameter. The OVRTAPF command can also specify parameters to override values contained in the file description of the overriding file. If the file named in the program is not replaced, but certain parameters of the file are overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters overridden are then specified by the other parameters of the OVRTAPF command. Parameters that are not specified do not affect the parameters specified in the file description, in the program, or in other called file override commands.


Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>File being overridden</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>TOFILE</td>
<td>Overriding to tape file</td>
<td>Single values: *FILE</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Overriding to tape file</td>
<td>Name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td>Positional 3</td>
</tr>
<tr>
<td>DEV</td>
<td>Device</td>
<td>Values (up to 4 repetitions): Name</td>
<td>Optional</td>
</tr>
<tr>
<td>VOL</td>
<td>Volume identifier</td>
<td>Single values: *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values (up to 50 repetitions): Character value</td>
<td></td>
</tr>
<tr>
<td>REELS</td>
<td>Tape reels specifications</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 2: Number of reels</td>
<td>1-255</td>
<td></td>
</tr>
<tr>
<td>SEQNBR</td>
<td>Sequence number</td>
<td>1-16777215, *END, *NEXT</td>
<td>Optional</td>
</tr>
<tr>
<td>LABEL</td>
<td>File label</td>
<td>Character value</td>
<td>Optional</td>
</tr>
<tr>
<td>RCDLLEN</td>
<td>Record length</td>
<td>Integer, *CALC</td>
<td>Optional</td>
</tr>
</tbody>
</table>
### File being overridden (FILE)

Specifies the file being used by the program to which this override command is applied. The specified file must be a tape device file when *FILE is specified in the Overriding to tape file (TOFILE) parameter. Otherwise, any device file or database file name can be specified.

This is a required parameter.

**name** Specify the name of the file.
Overriding to tape file (TOFILE)

 specifies the tape file that is used instead of the file specified in the File being overridden (FILE) parameter; or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified on this command override the other values specified in the tape device file or in the program.

**Single values**

*FILE*  The tape device file named on the FILE parameter has some of its parameters overridden by values specified in this command.

**Qualifier 1: Overriding to tape file**

*name*  Specify the name of the tape device file that is used instead of the overridden 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 tape device file. If no library is specified as the current library for the job, QGPL is used.

*name*  Specify the library where the tape device file is located.

Device (DEV)

specifies the names of up to four tape devices, one virtual tape device, or one media library device that are used with the tape device file to perform input/output operations. A media library device is a tape storage device that contains one or more tape drives, tape cartridges, and a part (carriage and picker assembly) for moving tape media between the cartridge storage slots and the tape drives. The order in which the device names are specified here is the order in which tapes placed in the devices are processed. Specify the device names (no more than four) that override the device names specified in the program or in the tape device file. When the number of volumes processed exceeds the number of devices in the DEV list, the devices are used in the same order as specified, wrapping around to the first device as needed.

*name*  Specify the name of the tape device or media library device.

Volume identifier (VOL)

specifies one or more volume identifiers of the tapes that are used by the tape device file. The tapes (volumes) must be written on the devices in the same order as their identifiers are specified here, and in the same order as the device names are specified on the Device (DEV) parameter. If the tape file is opened for read backward, the volume identifiers in the list are processed from last to first, while the devices in the device list are used in first to last order.

**Single values**

*NONE*  No tape volume identifiers are specified for this file. They are supplied before the device file is opened, either in a CHGTAPF or in another Override with Tape File (OVRTAPF) command. If no
volume identifiers are specified before the device file is opened, no volume checking is performed beyond verifying that the correct label type volume is put on the device, and no volume names are provided in operator messages.

Other values (up to 50 repetitions)

**character-value**

Specify the identifiers of one or more volumes in the order in which they are placed on the device. Each volume identifier contains a maximum of 6 alphanumeric characters. Use a blank as a separator character when listing multiple identifiers. Up to 50 volume identifiers can be specified. These identifiers are used in messages sent to the operator during processing. The maximum number of reels processed for an *NL, *NS, *BLP, or *LTM input file is determined by the number of volume identifiers in the list.

**Note:** If the VOL parameter value used for the file specifies a list of identifiers rather than VOL(*NONE), the number-of-reels part of the REELS parameter is ignored regardless of where it is specified. A description of how the parameter values for the file are determined when overrides are used, the high-level language interface, and the device file when the file is opened is in the Files and file systems topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter. To ensure that the number-of-reels part of the REELS parameter is used (rather than a VOL identifier list) to control the volumes processed by the tape device file, specify VOL(*NONE) in the same command in which the REELS parameter is specified.

---

**Tape reels specifications (REELS)**

Specifies the type of labeling that is used on the tape reels, and the maximum number of reels processed if there is no list of volume identifiers specified on the **Volume identifier (VOL)** parameter and this device file is used with either *NL, *LTM, *NS, or *BLP input files. When the number of reels is specified, the volume identifiers of any labeled volumes placed in the tape device are ignored; instead, the order in which the reels are used must be checked by the operator.

The maximum number of reels specification (the second part of this parameter) is not a limiting value for standard-label input files or output files. For a standard-label input file, the data file labels limit the number of volumes processed by indicating end of file. For an output file, the maximum number of reels value is ignored; the system requests that additional volumes be placed in the device until the file is closed.

The system checks the first record following the load point on the tape to see (1) whether it has exactly 80 bytes for EBCDIC or at least 80 bytes for ASCII and (2) whether the first 4 bytes contain the values VOL and 1. If so, the reel contains a standard-label tape. *SL and *BLP files require standard-label tape volumes. *NL, *NS, and *LTM tape files cannot process standard-label volumes.

**Note:** The values *SL, *NL, and *LTM can be specified if the device file is used for either reading or writing on tapes. The values *NS and *BLP are valid only if the device file is used to read tapes.

This parameter overrides the values specified in the device file, in the program, or in other called OVRTAPF commands.

**Element 1: Label processing type**

*SL  The volumes have standard labels. If a list of volume identifiers is specified (with the VOL parameter), the system checks that the correct tape volumes are on the device in the specified sequence.
• If no volume identifier list is given and the file is opened for output, any standard-label volumes may be installed on the device.

• If no volume identifier list is given and the file is opened for input, the first volume may have any volume identifier, but if the file is continued, the system requires the correct continuation volumes to be processed (verified by checking the data file labels). For an input file, the end-of-file message is sent to the program being used when the labels on the last volume processed indicate that it is the last volume for the data file.

*NL The volumes are not labeled. On a nonlabeled volume, tape marks are used to indicate the end of each data file and the end of the volume. For an input file, the end-of-file message is sent to the program when the number of volumes specified in the volume list have been processed, or, if no list of volume identifiers is provided, when the number of reels specified in the REELS parameter are processed.

*NS The volumes have nonstandard labels. Each volume must start with some kind of label information, optionally preceded by a tape marker and always followed by a tape marker. This nonstandard label information is ignored. The system spaces forward to a point beyond the tape marker that follows the nonstandard labels and positions the tape at the file’s data. Each reel must have a tape marker at the end of the file’s data. Information beyond this ending tape marker is ignored. Only a single data file can exist on a nonstandard tape. Standard-label volumes cannot be processed by using the *NS label processing.

For an input file, the end-of-file message is sent to the program using the file when the number of volumes specified in the volume list have been processed, or, if no list of volume identifiers is provided, when the number of reels specified in the REELS parameter are processed.

*BLP Standard-label processing is bypassed. Each reel must have standard labels. Although each reel is checked for a standard volume label and each file must have at least one standard header label (HDR1) and one standard trailer label (EOV1 or EOF1), most other label information (such as the data file record length or block length) is ignored. The sequence number of each file on the volume is determined only by the number of tape markers between it and the start of tape (in contrast to *SL processing in which the file sequence number stored in the header and trailer labels of each file are used to locate a data file).

Most of the information in the data file trailer label is ignored, but if an end-of-file (EOF) trailer label is found, the end-of-file message is sent to the program using the tape file. If no end-of-file trailer label is encountered by the time the specified number of volumes or reels have been processed (volume identifier list and REELS parameter), the end-of-file message is immediately sent to the program using the tape file. Bypass label processing can be used when the user does not know the name of the file used or when some file label information is incorrect.

*LTM The volumes have no labels but do have a single leading tape marker before the first data file. REELS(*LTM) is processed the same as REELS(*NL) except that when SEQNBR(1) is specified for an output file to create the first data file on the tape, a leading tape marker is written at the start of the tape before the first data block.

Element 2: Number of reels

1-255 Specify the maximum number of reels that are processed for an *NL, *LTM, *NS, or *BLP input tape operation when there is no list of volume identifiers used on the Volume identifier (VOL) parameter. If the next reel is not on the device when the end of the currently-processing tape is reached, a message is sent to the operator requesting that the next tape be installed on the next tape device. The number-of-reels value is ignored for a standard label (*SL) processing file, or for any output file.
Sequence number (SEQNBR)

Specifies the sequence number of the data file on the tape that is processed.

- When standard-label tapes are used, the four-position file sequence number is read from the first header label of the data file.
- When bypass label processing is used, or when standard-label tapes are not used, the system counts the tape marks from the beginning of the tape to locate the data file with the correct sequence number.
- When multiple file, multiple volume tapes are processed using *SL on the Tape reels specifications (REELS) parameter, the file sequence numbers continue consecutively through all of the volumes; each new data file has a sequence number that is one greater than the previous file, regardless of which volume contains the file.

I-16777215

Specify the file sequence number that overrides the sequence number specified in the program or device file.

*END The file sequence number is added to the end of the tape.

An error message is shown on the display when a tape device file is used to read from a tape and the *END special value is specified in the tape device file.

*NEXT

The next file on the tape is processed. If the tape is currently positioned before the first file, the first file on the tape is processed. This value can only be specified in tape files that are used to read from tape. An error message is issued when a tape file is used to write to a tape and *NEXT is specified in the tape file.

File label (LABEL)

Specifies the identifier of the data file that is processed by this tape device file. A label identifier is required for a standard label output file, but is optional for an input file.

If a data file identifier is specified for any type of label processing other than *SL, it is ignored.

character-value

Specify the tape data file identifier.

Record length (RCDLEN)

Specifies (in bytes) the length of the records that are contained in the data file that is processed with this device file. The system always uses the record length and block length specified in the data file labels for any standard label input file or output file with *YES specified in the Extend file (EXTEND) parameter, if a second header label (HDR2) is found on the tape and *BLP label processing is not specified. This parameter overrides the value specified in the device file, in the program, or in other called OVRTAPF commands.

*CALC

No record length is specified for the data file processed. If *CALC is specified, the system attempts to calculate an appropriate record length when the file is opened. *CALC is used for tapes that are not labeled or when there is no HDR2 label if a value other than *CALC is specified in the Block length (BLKLEN) parameter for the file, and if the Record block format (RCDBLKFM) parameter does not specify spanned or blocked records. In this case, the system calculates an appropriate record length from the block length, record block format, and buffer
offset (for an ASCII file) specified for the file. In any other case, the actual record length must be specified by a Change Tape File (CHGTAPF) command or Override with Tape File (OVRTAPF) command, or in the high-level language program that opens the device file.

integer
Specify a value ranging from 1 through 32767 that specifies the length of each record in the data file. The minimum and maximum record lengths that are allowed for a file are dependent on the record block format, block length, buffer offset (for an ASCII file), and recording code.

Block length (BLKLEN)
Specifies (in bytes) the maximum length of the data blocks transferred to or from the tape for input or output operations. The system always uses the block length and record length specified in the data file labels for any standard label input file or output file with *YES specified in the Extend file (EXTEND) parameter, if a second header label (HDR2) is found on the tape and *BLP label processing has not been specified.

This parameter overrides the value specified in the device file, in the program, or in other OVRTAPF commands.

*CALC
No block length is specified for the data file processed. If *CALC is specified, the system attempts to calculate an appropriate block length when the file is opened. *CALC can be used for tapes that are not labeled or when there is no HDR2 label if a value other than *CALC is specified in the Record length (RCDLEN) parameter for the file, and if the Record block format (RCDBLKfmt) parameter does not specify spanned or blocked records. In this case, the system calculates an appropriate block length from the record length, record block format, and buffer offset (for an ASCII file) specified for the file. In any other case, the actual block length must be specified by a Change Tape File (CHGTAPF) command or Override with Tape File (OVRTAPF) command, or in the high-level language program that opens the device file.

1-524288
Specify the maximum length, in bytes, of each block in the data file to be processed. The minimum block length that can be successfully processed is determined by the tape device hardware and AS/400 system machine support functions.

The maximum block length is always 524288 bytes for an input file, but is limited to 9999 bytes if block descriptors must be created for an ASCII output file.

The following table shows the minimum and maximum block length values allowed for an output file:

<table>
<thead>
<tr>
<th>CODE</th>
<th>BUFOFSET</th>
<th>MIN BLKLEN</th>
<th>MAX BLKLEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>*EBCDIC</td>
<td>Ignored</td>
<td>18</td>
<td>524288</td>
</tr>
<tr>
<td>*ASCII</td>
<td>0</td>
<td>18</td>
<td>524288</td>
</tr>
<tr>
<td>*ASCII</td>
<td>*BLKDSY</td>
<td>18</td>
<td>9999</td>
</tr>
</tbody>
</table>

Buffer offset (BUFOFSET)
Specifies the buffer offset value for the start of the first record in each block in the tape data file. A buffer offset value is used for any record block format ASCII file, and is ignored for an EBCDIC tape file. The system always uses the buffer offset specified in the data file labels for any standard label input file or output file with *YES specified in the Extend file (EXTEND) parameter, if a value is contained in the second header label (HDR2) on the tape, and *BLP label processing is not specified.
The buffer offset parameter specifies the length of any information that precedes the first record in the block. For record block formats *D, *DB, *VS, and *VBS, each record or record segment is preceded by a descriptor that indicates the length of the record or segment. A buffer offset value is used to indicate that there is information ahead of the descriptor word for the first record in each block, or ahead of the data of the first fixed-length record or undefined format record in each block.

This parameter is not needed for a standard-labeled file that is processed for input if the tape includes a second file header label (HDR2) that contains the buffer offset value. A buffer offset must be provided by the Create Tape File (CRTTAPF) command, Change Tape File (CHGTAPF) command, or Override with Tape File (OVRTAPF) command, or by the file labels for an input file that contains any information (such as a block descriptor) ahead of the first record in each block. If you do not specify a buffer offset when a tape file is created, it is not necessary to specify an offset value when the file is read.

The only buffer offset values allowed for an output file are zero and *BLKDSC. An existing standard-labeled data file with a buffer offset value in the HDR2 label is extended only if the offset value is either 0 or 4. An offset value of 0 in the HDR2 label adds data blocks with no buffer offset. *BLKDSC must be specified to extend an existing tape data file that contains an offset value of 4 in the HDR2 label.

This parameter overrides the value specified in the device file, in the program, or in other called OVRTAPF commands.

*BLKDSC

Block descriptors that are 4 bytes in length are created in any tape file that is created using this device file. Any input file read using this device file assumes 4 bytes of buffer offset information preceding the first record in each data block. This value is valid only for a record block format *D or *DB file. The contents of the buffer offset information of each output data block when BUFOFSET(*BLKDSC) is specified is the actual length of the data block, expressed in zoned decimal format.

integer

Specify a value ranging from zero through 99 that specifies the length of the buffer offset information that precedes the first record in each data block.

---

**Record block format (RCDBLKFMT)**

Specifies the type of format blocking attribute of records in the tape data file being processed.

Record block format *V and *VB records can be processed only for an EBCDIC file; *D and *DB records can be processed only for an ASCII file. If a standard-label tape (label type *SL or *BLP) is being processed and an inconsistent record block format is specified for the volume code, the correct record type is assumed (V or D) for the volume code and a warning message is sent to the program that opens the file. If the record type and code are inconsistent for a nonlabeled volume (label type *NL, *LTM, or *NS), an error message is sent and the file is not opened, because there are no labels to verify the correct volume code.

If a valid record length, block length, and buffer offset value (for an ASCII file) are specified for fixed-length records but the block attribute is incorrect, the correct block attribute is assumed (changing record block format *F to *FB or record block format *FB to *F), and a warning message is sent to the program that opens the file.

If a block length is specified that is longer than required to process a maximum length record, then record block format *V, *D, or *VS is changed to *VB, *DB, or *VBS and a warning message is sent to the program that opens the file.
**Note:** When BUFOSET(*BLKDSC) is specified for the file, a value of 4 should be used for the BUFOSET part of any BLKLEN calculations, unless existing file labels on the tape specify a different value.

This parameter overrides the value specified in the device file, in the program, or in other called OVRTAPF commands.

*F  Fixed length, deblocked, unspanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *FB, based on other file parameters.

*FB Fixed length, blocked, unspanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *F, based on other file parameters.

*V Variable length, deblocked, records in EBCDIC type V format are processed. The system may change this record block format to *VB, *D, or *DB, based on other file parameters.

*VB Variable length, blocked, unspanned records in EBCDIC type V formats are processed. The system may change this record block format to *DB, based on the volume code.

*D Variable length, deblocked, unspanned records in ASCII type D formats are processed. The system may change this record block format to *DB, *V, or *VB, based on other file parameters.

*DB Variable length, blocked, unspanned records in ASCII type D formats are processed. The system may change this record block format to *VB, based on the volume code.

*VS Variable length, deblocked, spanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *VBS, based on other file parameters. The representation of spanned records on the tape is different for EBCDIC and ASCII files, but the system selects the correct format based on the file code.

*VBS Variable length, blocked, spanned records in either EBCDIC or ASCII code are processed. The representation of spanned records on the tape is different for EBCDIC and ASCII files, but the system selects the correct format based on the file code.

*U Undefined format records in either EBCDIC or ASCII code are processed. Records are processed as variable length records, where each record written or read is in a separate tape block.

---

**Extend file (EXTEND)**

Specifies, for output operations to tape, whether new records are added to the end of a data file that is currently on the tape. The specific data file is identified by the Sequence number (SEQNBR) parameter and, for a standard-label file, by the File label (LABEL) parameter. If the data file is extended, it becomes the last file on the tape volume. Any data files that follow this data file are overwritten as the specified file is extended.

This parameter overrides the extend value specified in the device file, in the program, or in other called OVRTAPF commands.

**Single values**

*NO Records are not added to the end of the specified data file. Regardless of whether there is already a data file with the specified sequence number on the tape, a new data file is created that overwrites an existing data file and any files that follow it.

**Element 1: Extend file**

*YES New records are added to the end of the specified data file.

**Element 2: Check file**
*NOCHECK
   The file is extended without being checked to determine whether it is active.

*CHECK
   Before the file is extended, it is checked to determine whether it is active.

---

**Tape density (DENSITY)**

Specifies the density of the data that is written on the tape volume when this device file is created. This parameter is used only for tape files being written to tape; it is ignored for tape files being read from the tape (in the case of files being read from tape, the density on the tape is used).

The density of a standard-label volume is specified on the INZTAP command, which initializes tapes as standard-label volumes by writing volume labels on them. If the density specified on this parameter is different than the density of a standard-labeled tape, the tape must be reinitialized to the specified density.

**DEVTYPE**
   The highest capacity density or format supported by the tape device will be used.

<table>
<thead>
<tr>
<th>Device</th>
<th>Highest capacity density or format</th>
</tr>
</thead>
<tbody>
<tr>
<td>3480</td>
<td>*FMT3480</td>
</tr>
<tr>
<td>3490E</td>
<td>*FMT3490E</td>
</tr>
<tr>
<td>3570-Bxx</td>
<td>*FMT3570</td>
</tr>
<tr>
<td>3570-Cxx</td>
<td>*FMT3570E</td>
</tr>
<tr>
<td>3580-001</td>
<td>*ULTRIUM1</td>
</tr>
<tr>
<td>3580-002</td>
<td>*ULTRIUM2</td>
</tr>
<tr>
<td>3580-003</td>
<td>*ULTRIUM3</td>
</tr>
<tr>
<td>3590-Bxx</td>
<td>*FMT3590</td>
</tr>
<tr>
<td>3590-Exx</td>
<td>*FMT3590E</td>
</tr>
<tr>
<td>3590-Hxx</td>
<td>*FMT3590H</td>
</tr>
<tr>
<td>3592-J1A</td>
<td>*FMT3592A1</td>
</tr>
<tr>
<td>4685-001</td>
<td>*VXA2</td>
</tr>
<tr>
<td>6335</td>
<td>*QIC3040</td>
</tr>
<tr>
<td>6343</td>
<td>*QIC1000</td>
</tr>
</tbody>
</table>
The highest capacity density or format supported by the device for the mounted cartridge type will be used. If the device does not support special cartridge type information, *DEVTYPE is used.

**character-value**
Specify the density or format to use.

- **1600** The data density on the tape volume is 1,600 bits per inch, which is used for 1/2 inch reel tapes.
- **3200** The data density on the tape volume is 3,200 bits per inch, which is used for 1/2 inch reel tapes.
- **6250** The data density on the tape volume is 6,250 bits per inch, which is used for 1/2 inch reel tapes.
*FMT3480
The format of this tape is FMT3480. The data density on this tape volume is formatted to support a 3480 device. This density is used for 1/2 inch cartridge tapes.

*FMT3490E
The format of this tape is FMT3490E. The data density on this tape volume is formatted to support a 3490E device. This density is used for 1/2 inch cartridge tapes.

*FMT3570
The format of this tape is FMT3570. The data format is written on the tape volume with a 3570 device.

*FMT3570E
The format of this tape is FMT3570E. The data format is written on the tape volume with a 3570E device.

*FMT3590
The format of this tape is FMT3590. The data format is written on the tape volume with a 3590 device. This density is used for 1/2 inch cartridge tapes.

*FMT3590E
The format of this tape is FMT3590E. The data format is written on the tape volume with a 3590E device. This density is used for 1/2 inch cartridge tapes.

*FMT3590H
The format of this tape is FMT3590H. The data format is written on the tape volume with a 3590H device. This density is used for 1/2 inch cartridge tapes.

*FMT3592A1
The format of this tape is FMT3592A1. The data format is written on the tape volume with a 3592 device. This density is used for 1/2 inch cartridge tapes.

*QIC120
The format of this tape is QIC120, which is used for 1/4 inch cartridge tapes that can hold 120 megabytes of data.

*QIC525
The format of this tape is QIC525, which is used for 1/4 inch cartridge tapes that can hold 525 megabytes of data.

*QIC1000
The format of this tape is QIC1000, which is used for 1/4 inch cartridge tapes that can hold 1200 megabytes of data.

*QIC2GB
The format of this tape is QIC2GB. It is used by 1/4 inch tape devices which can store 2.5 gigabytes of data on a standard length QIC2GB cartridge.

*QIC2DC
The format of this tape is QIC2DC. It is used to write compacted data to a 1/4 inch cartridge that supports the QIC2GB format.

*QIC4GB
The format of this tape is QIC4GB. It is used by 1/4 inch tape devices which can store 4 gigabytes of data on a standard length QIC4GB cartridge.

*QIC4DC
The format of this tape is QIC4DC. It is used to write compacted data to a 1/4 inch cartridge that supports the QIC4GB format.

*QIC3040
The format of this tape is QIC3040, which is used for 1/4 inch minicartridge tapes that can hold 840 megabytes of data.
*QIC5010
  The format of this tape is QIC5010, which is used for 1/4 inch cartridge tapes that can hold 13.5 gigabytes of data.

*MLR3
  The format of this tape is MLR3. It is used by 1/4 inch tape devices which can store 25 gigabytes of data on a standard length MLR3 cartridge.

*SLR60
  The format of this tape is SLR60. It is used by 1/4 inch tape devices which can typically store 60 gigabytes of compacted data on a standard length SLR60 cartridge.

*SLR100
  The format of this tape is SLR100. It is used by 1/4 inch tape devices which can typically store 100 gigabytes of compacted data on a standard length SLR100 cartridge.

*FMT2GB
  The format of this tape is FMT2GB, which is used for 8 millimeter cartridge tapes that can hold 2 gigabytes of data.

*FMT5GB
  The format of this tape is FMT5GB, which is used for 8 millimeter cartridge tapes that can hold 5 gigabytes of data.

*FMT7GB
  The format of this tape is FMT7GB, which is used for 8 millimeter cartridge tapes that can hold 7 gigabytes of data.

*FMT20GB
  The format of this tape is FMT20GB. It is used by 8 millimeter tape devices that can store 20 gigabytes of data on a standard length cartridge.

*FMT60GB
  The format of this tape is FMT60GB. It is used by 8 millimeter tape devices that can store 60 gigabytes of data on a standard length cartridge.

*ULTRIUM1
  The format of this tape is ULTRIUM1. It is used by 1/2 inch cartridge tape devices that can store 100 gigabytes of data on a standard length cartridge.

*ULTRIUM2
  The format of this tape is ULTRIUM2. It is used by 1/2 inch cartridge tape devices that can store 200 gigabytes of data on a standard length cartridge.

*ULTRIUM3
  The format of this tape is ULTRIUM3. It is used by 1/2 inch cartridge tape devices that can store 400 gigabytes of data on a standard length cartridge.

*VRT32K
  The format of the volume is VRT32K. It is used to write data to a virtual volume using a maximum data block size of 32KB. Volumes written using this format can be duplicated to all supported tape devices.

*VRT64K
  The format of the volume is VRT64K. It is used to write data to a virtual volume using a maximum data block size of 64KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 64KB or greater.

*VRT240K
  The format of the volume is VRT240K. It is used to write data to a virtual volume using a maximum data block size of 240KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 240KB or greater.
*VRT256K
The format of the volume is VRT256K. It is used to write data to a virtual volume using a maximum data block size of 256KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 256KB or greater.

*VXA1
The format of this tape is VXA1. It is used by VXA cartridge tape devices that can store 33 gigabytes of data on a standard length cartridge.

*VXA2
The format of this tape is VXA2. It is used by VXA cartridge tape devices that can store 80 gigabytes of data on a standard length cartridge.

Note: Self-configured tape devices may define additional valid values for the density parameter. Use iSeries Navigator (Configuration and Service) (Hardware) (Tape Devices) (Tape Libraries) (Tape Resources) (Properties) or (Configuration and Service)(Hardware) (Tape Devices) (Stand-Alone Devices) (Properties) to find additional valid density values for a specific device, or use the F4=Prompt key on the "Tape density" field of the CL command to see a list of all valid density values for the attached tape devices.

Data compaction (COMPACT)
Specifies whether device data compaction is performed. If the tape devices being used do not support data compaction, this parameter will be ignored when the file is opened.

This parameter overrides the value specified in the device file, in the program or in other called OVRTAPF commands.

*DEVD
Device data compaction is performed if the devices being used support data compaction.

*NO
Device data compaction is not performed.

Code (CODE)
Specifies the type of character code that is used by the tape device file when the system is reading or writing tape data.

*EBCDIC
The EBCDIC character code is used with this tape device file.

*ASCII
The ASCII character code is used.

Note: For standard labeled (*SL) tapes the CODE parameter is used to determine how the labels are processed. For all label types the TBL, FROMCCSID, and TOCCSID parameters control what conversion, if any, is used for the data portion of the files.
Creation date (CRTDATE)
Specifies, for tape input data files and for tape output for which *YES is specified in the Extend file (EXTEND) parameter, the date when the data file was written to tape. The creation date of the data file is stored in file labels on the tape. If a creation date is specified for any type of label processing other than *SL, it is ignored.

This parameter overrides the value specified in the program, device file, or in other called OVRTAPF commands.

*NONE
The creation date of the data file is not checked.

date Specify the creation date of the data file. The date must be specified in job-date format.

File expiration date (EXPDATE)
Specifies, for tape output data files, the expiration date of the data file used by this device file. The data file expiration date is stored in file labels on the tape. If an expiration date is specified for any type of label processing other than *SL, it is ignored. If a date is specified, the data file is protected and cannot be overwritten until the specified expiration date.

This parameter overrides the value specified in the program, device file, or in other called OVRTAPF commands.

*NONE
No expiration date for the data file is specified; the file is not protected. An expiration date is written in the data file labels so the file can be used as a scratch data file.

*PERM
The data file is protected permanently. The date written in the tape data file is 999999.

date Specify the date on which the data file expires. The date must be specified in job-date format.

End of tape option (ENDOPT)
Specifies the operation that is automatically performed on the tape volume after the operation ends. If more than one volume is included, this parameter applies only to the last tape volume used; all other tape volumes are rewound and unloaded when the end of the tape is reached.

*RENEW
The tape is rewound, but not unloaded.

*UNLOAD
The tape is automatically rewound and unloaded after the operation ends.

*LEAVE
The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.
User label program (USRLBLPGM)

Specifies that a program processes user header and trailer labels on a standard-labeled tape. This parameter is valid only when *SL is specified in the Tape reels specifications (REELS) parameter.

Single values

*NONE

Only standard label processing is used. No program is called to process user labels.

Qualifier 1: User label program

name Specify the name of the program that is called to process user labels.

Qualifier 2: Library

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

*CURLIB

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

name Specify the library where the program is located.

Conversion table (TBL)

Specifies the qualified name of a conversion table to be used for single-byte conversion of input files or output files. The specified conversion is only used for the data portion of the files. When the specified code is *ASCII (CODE parameter) any labels will be converted between ISO/ASCII 8-Bit code and EBCDIC. When the specified code is *EBCDIC (CODE parameter) the labels, if any, are not converted.

Note: See system supplied conversion tables QSYS/QASCII and QSYS/QEBCDIC for an example of the conversion used to translate between ISO/ASCII 8-Bit code and EBCDIC.

Single values

*DFT When the specified code is *ASCII (CODE parameter) the data and labels will be converted between ISO/ASCII 8-bit code and EBCDIC. When the specified code is *EBCDIC (CODE parameter) the data and labels will not be converted.

*NONE

The data will not be converted.

*CCSID

The CCSID parameters are used to generate a conversion table to use for converting the data portion of the files.

Qualifier 1: Conversion table

name Specify the name of a conversion table to be used for conversion of the data between single-byte character sets.

Qualifier 2: Library

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

*CURLIB

The current library for the job is used to locate the program. If no library is specified as the current library for the job, QGPL is used.
name Specify the name of the library to be searched.

From CCSID (FROMCCSID)
Specifies a single-byte CCSID used for the input data. The input data is the data read from the tape for input operations, or read from a file for output operations.

1-65533 The requested CCSID value is used. The value is validated to ensure that a single-byte CCSID is specified.

To CCSID (TOCCSID)
Specifies the single-byte CCSID used for the output data. The output data is the data written to the tape for output operations, or written to a file for input operations.

1-65533 The requested CCSID value is used. The value is validated to ensure that a single-byte CCSID is specified.

User specified DBCS data (IGCDTA)
Specifies whether the file processes double-byte character set (DBCS) data.

*YES For program described files, indicates that the file is used to process DBCS data. For field level files, the fields that are defined with the ALTTYP keyword are changed from ALPHA fields or character fields to DBCS fields.

*NO For program described files, indicates that the file is not used to process DBCS data. For field level files, the fields that are defined with the ALTTYP keyword remain ALPHA fields or character fields. The file attribute remains at whatever it was set to when the file was created.

Maximum file wait time (WAITFILE)
Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

*IMMED The program does not wait. Immediate allocation of file resources is required.

*CLS The default wait time specified in the class description is used as the wait time for the file resources to be allocated.

integer Specify the number of seconds that the program waits for the file resources to be allocated to the tape file when the file is opened, or the wait time for the device allocated when an acquire operation is performed to the file. Valid values range from 1 through 32767 seconds.
Secure from other overrides (SECURE)

Specifies whether this file is safe from the effects of file override commands started in previously called programs.

*NO  This file is not protected from other file overrides; its values are overridden by the effects of any file override commands started in previously called programs.

*YES  This file is protected from the effects of any file override commands started in previously called programs.

Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.

*ACTGRPDFN

The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

*CALLLVL

The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

*JOB  The scope of the override is the job in which the override occurs.

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

*NO  An ODP created for this file open operation is not shared. Every time a program opens the file, a new ODP to the file is created and started.

*YES  If the file is opened more than once, the same ODP is shared with each program in the routing step that also specifies *YES for this parameter when it opens the file. This includes multiple open operations in the same program.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

*ACTGRPDFN

The scope of the open operation is determined by the activation group of the program that called the OVRTAPF command processing program. If the activation group is the default activation
group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

*JOB The scope of the open operation is the job in which the open operation occurs.

Examples

Example 1: Overriding a File

OVRTAPF FILE(OUT) VOL(DPT706) LABEL(STATUSR)

This command overrides a file named OUT in the program using the data file STATUSR on tape volume DPT706.

Example 2: Allowing DBCS Data

OVRTAPF FILE(IGCLIB/IGCTAP) IGCDTA(*YES)

This command overrides the tape device file named IGCTAP, which is stored in the library IGCLIB, so the file may contain double-byte character set data.

Example 3: Using Data Density of 1600 Bits Per Inch

OVRTAPF FILE(OUT) DENSITY(1600)

This command overrides a file named OUT to use a data density of 1600 bits per inch when writing to the tape volume.

Example 4: Using a Conversion Table to Process a Tape with EBCDIC Labels.

OVRTAPF FILE(FILE1) REELS(*SL) CODE(*EBCDIC)
TBL(LIB1/TABLE1)

This command overrides a tape device file named FILE1 to specify that a conversion table named LIB1/TABLE1 is to be used to convert all data read from, or written to, the tape volume.

Example 5: Using Specified CCSIDs to Process a Non-labeled Tape.

OVRTAPF FILE(FILE2) REELS(*NL) TBL(*CCSID)
FROMCCSID(819) TOCCSID(37)

This command overrides a tape device file named FILE2 to specify that any data read from, or written to, the tape volume is to be converted from CCSID 819 to CCSID 37.

Error messages

*ESCAPE Messages

CPF180C
Function &1 not allowed.

CPF1892
Function &1 not allowed.
Parameter Definition (PARM)

The Parameter (PARM) command definition statement defines a parameter of a command being created. A parameter is the means by which a value is passed to the command processing program. One PARM statement must be used for each parameter that appears in the command being defined. The order in which the PARM statements are entered into the source file determines the order in which the parameters must be specified when the command is entered in positional form and the order in which they are passed to the validity checker and the command processing program. A maximum of 99 parameters can be defined for one command. A command with a large number of parameters will require more processing time before the command processing program is called, regardless of how many parameters are actually coded.

Note: The PARM statement contains certain parameters and predefined values that can be used only when IBM-supplied command processing programs are called by the command being defined. Because of limitations in some high-level languages, these values may not be useful in the definition statements of user-defined commands. These parameters and values are identified by the phrase (For IBM-supplied commands) that immediately follows the parameter keyword (if the entire parameter is for IBM-supplied commands only) or the predefined value to which it applies.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>KWD</td>
<td>Keyword</td>
<td>Simple name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>LEN</td>
<td>Value length</td>
<td>Values (up to 3 repetitions): Integer</td>
<td>Optional, Positional 3</td>
</tr>
<tr>
<td>RTNVAL</td>
<td>Return value</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>CONST</td>
<td>Constant value</td>
<td>Character value</td>
<td>Optional</td>
</tr>
<tr>
<td>RSTD</td>
<td>Restricted values</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>DFT</td>
<td>Default value</td>
<td>Character value</td>
<td>Optional</td>
</tr>
<tr>
<td>VALUES</td>
<td>Valid values</td>
<td>Values (up to 300 repetitions): Character value</td>
<td>Optional</td>
</tr>
<tr>
<td>REL</td>
<td>Relational expression</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td>RANGE</td>
<td>Range of values</td>
<td>Element list</td>
<td>Optional</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 1998, 2006
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPCVAL</td>
<td>Special values</td>
<td>Values (up to 300 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: From value</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: To replacement value</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td>SNGVAL</td>
<td>Single values</td>
<td>Values (up to 300 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: From value</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: To replacement value</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td>Minimum values required</td>
<td>0-300, 0</td>
<td>Optional</td>
</tr>
<tr>
<td>MAX</td>
<td>Maximum values allowed</td>
<td>Integer, 1</td>
<td>Optional</td>
</tr>
<tr>
<td>ALWUNPRT</td>
<td>Allow unprintable characters</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>ALVVAR</td>
<td>Allow variable names</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>PGM</td>
<td>Is PARM a program name</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>DTAARA</td>
<td>Is PARM a data area name</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>FILE</td>
<td>If a file parameter, how used</td>
<td>*NO, *IN, *OUT, *UPD, *INOUT, *UNSPFD</td>
<td>Optional</td>
</tr>
<tr>
<td>FULL</td>
<td>Full field required</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>EXPR</td>
<td>Value an expression</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>VARY</td>
<td>Varying length</td>
<td>Single values: *NO Other values: Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Return length value</td>
<td>*YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Value length</td>
<td>*INT2, *INT4</td>
<td></td>
</tr>
<tr>
<td>PASSATR</td>
<td>Pass attribute byte</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>PASSVAL</td>
<td>Value to pass if unspecified</td>
<td>*DFT, *NULL</td>
<td>Optional</td>
</tr>
<tr>
<td>CASE</td>
<td>Case of value</td>
<td>*MONO, *MIXED</td>
<td>Optional</td>
</tr>
<tr>
<td>CCSID</td>
<td>CCSID of value</td>
<td>*JOB, *UTF16</td>
<td>Optional</td>
</tr>
<tr>
<td>LISTDSPL</td>
<td>List displacement</td>
<td>*INT2, *INT4</td>
<td>Optional</td>
</tr>
<tr>
<td>DSPINPUT</td>
<td>Display input</td>
<td>*YES, *PROMPT, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>CHOICE</td>
<td>Choice text</td>
<td>Character value, *VALUES, *NONE, *PGM</td>
<td>Optional</td>
</tr>
<tr>
<td>CHOICEPGM</td>
<td>Choice program</td>
<td>Single values: *NONE Other values: Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Choice program</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>PMTCTL</td>
<td>Prompt control</td>
<td>Simple name, *NONE, *PMTRQS</td>
<td>Optional</td>
</tr>
<tr>
<td>PMTCTLPGM</td>
<td>Prompt control program</td>
<td>Single values: *NONE Other values: Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Prompt control program</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>KEYPARM</td>
<td>Key parameter</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>INLPMTLEN</td>
<td>Initial prompt length</td>
<td>*CALC, *PWD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 17, 25, 32, 50, 80, 132, 256, 512</td>
<td>Optional</td>
</tr>
<tr>
<td>PROMPT</td>
<td>Prompt specifications</td>
<td>Single values: *NONE Other values: Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Prompt text or message ID</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Order prompt is displayed</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>
Keyword (KWD)

Specifies the keyword name of the parameter being defined. Every CL command parameter has an associated keyword name. Command parameters specified using the parameter keyword can be specified in any order. Parameters may be specified in positional form (without a keyword name) up to the positional limit specified by the MAXPOS parameter on the Create Command (CRTCMD) command.

`simple-name`

Specify a keyword name for the parameter. The name can be up to ten alphanumeric characters. The first character must be alphabetic.

Type of value (TYPE)

Specifies the type of the value that can be specified for the parameter named in KWD. The value can be an integer, a decimal, hexadecimal, or logical value, or a character string (optionally enclosed in apostrophes) that can be a name, date, or time. The value can also be a command. Type one of the following options to specify the parameter type:

*DEC  The parameter value is a packed decimal number.
*LGL  The parameter value is a logical value of one ('1') or zero ('0').
*CHAR  The parameter value is a character string that can optionally be enclosed in apostrophes. If the character string contains any special characters (not including an asterisk (*)), it must be enclosed in apostrophes. The maximum length of the character string is 5000 characters.
*NAME  The parameter value is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic or one of the special characters, $, @, or #. The remaining characters can be alphanumeric, a period, an underscore, or one of special characters, $, @, or #. The name can also be a string of characters starting and ending with double quotation marks (") or enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it should be specified on the Special values (SPCVAL) parameter.
*SNAME  The parameter value is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic or one of the special characters $, @, or #. The remaining characters can be alphanumeric, an underscore, or one of the special characters $, @, or #). The character string can be enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it must be specified on the Special values (SPCVAL) parameter.
*CNAME  The parameter value is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic or one of the special characters $, @, or #. The remaining characters can be alphanumeric or one of special characters, $, @, or #. The character string can be enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it must be specified on the Special values (SPCVAL) parameter.
*PNAME  The parameter value is a character string that represents a path name string. Optionally the path name string may be enclosed in apostrophes. If the path name string contains any special characters (not including an asterisk (*)), it must be enclosed in apostrophes. The maximum length of the path name string is 5000 characters.
*GENERIC

The parameter value is a character string that represents a generic name. A generic name contains a maximum of 255 characters followed by an asterisk (*) and must conform to the rules for generic names. The name identifies a group of objects whose names all begin with the characters preceding the asterisk (*). If an asterisk (*) is not included, the system assumes that the generic name is a complete object name.

*CMDSTR

The parameter value is a command that will be checked for validity by the system. It is passed to the command processing program as a command string.

The command analyzer rebuilds the command string when it checks it for validity. When the command is rebuilt, keywords are added to parameters that were specified positionally, parameters can be reordered, and parameters that contain characters that cannot be printed (X’FF’ and X’00 - X’3F’) are converted to hexadecimal notation. As a result, the rebuilt command string may be substantially longer than the original command string. If the length of the rebuilt command is longer than the allowed length specified with the LEN keyword, the command will fail.

Note: Selective prompting is not allowed with the *CMDSTR parameter.

*DATE

The parameter value is a character string that represents a date. When entering the command, the year may be specified with either 2 digits or 4 digits. If a 2-digit year is specified, the date is assumed to be in the range of January 1, 1940 through December 31, 2039. If a 4-digit year is specified, the date may be in the range of August 24, 1928 through May 9, 2071. When it is passed to the command processing program, it is always passed in the format Cyymmddd, where C = century, yy = year, mm = month, and dd = day. The century digit is set to 0 (zero) for years 19xx, and it is set to 1 (one) for years 20xx. When a date value is specified in this PARM statement, it must be specified in one of the following formats mmdddyy, mmddyyyy, or Cyymmddd. When a user types a date in the command at run time, it must be specified in the job-date format. The job date separator may be used when the date is entered. If the separator character is used, the date must be enclosed in apostrophes.

*TIME

The parameter value is a character string that represents a time. It is passed to the command processing program in a 6-byte character string as hhmmss, where hh = hours, mm = minutes, and ss = seconds. Values specified in this statement must be in the format hhmmss When a user types a time in the command at run time, it must be specified in the format hhmmss. The job time separator may be used when the time is entered. If the separator character is used, the time must be enclosed in apostrophes.

*HEX

The parameter value is hexadecimal in form. The specified characters must be 0 through F. They are converted to hexadecimal (EBCDIC) characters (2 hex digits per byte), right-justified, and padded on the left with zeros. If the value is enclosed in apostrophes, an even number of digits is required. If the value is not enclosed in apostrophes, an even number of digits is not required.

*ZEROELEM

The parameter is always considered as a list of zero elements, for which no value can be specified in the command. It is used to prevent a value from being entered for a parameter that is a list even though the command processing program expects one. For example, if two commands use the same command processing program, one command could pass a list for a parameter and the other command may not have any values to pass. The second command would be coded with *ZEROELEM specified for this parameter.

*X

(For IBM-supplied commands) The parameter value is a character string, variable name, or numeric value. The value is passed as a numeric value if it contains only digits, a + or - sign, or a decimal point; otherwise, it is passed as a character string.

*INT2

The parameter value is an integer that is passed as a 2-byte signed binary number.
*INT4  The parameter value is an integer that is passed as a 4-byte signed binary number.

*UINT2  The parameter value is an integer that is passed as a 2-byte unsigned binary number.

*UINT4  The parameter value is an integer that is passed as a 4-byte unsigned binary number.

*VARNAME  (For IBM-supplied commands) The parameter value is a CL variable name that is passed as a character string.

*CMD  (For IBM-supplied commands) The parameter value is a command. For example, the IF command has a parameter called THEN whose value must be another command. The command is checked for validity by the system.

*NULL  The parameter value is a null pointer, which can be used as a constant place-holder. A DEP statement or the REL and RANGE keywords of other PARM statements may not refer to the value of a parameter defined with *NULL specified for this parameter.

statement-label  Specify a qualified name or a mixed list of values. The statement label specified here is the statement label that identifies the first of a series of QUAL or ELEM statements that further describe the qualified name or the mixed list being defined by this PARM statement.

Value length (LEN)

Specifies the length of the parameter value that is passed to the command processing program. This parameter is not allowed if *INT2, *INT4, *UINT2, *UINT4, *DATE, *TIME, *CMD, *ZEROELEM, *NULL, or a statement label is specified for the Type of value (TYPE) parameter. With other values specified for the Type of value (TYPE) parameter, this parameter has the following applications:

- If *DEC is specified for the Type of value (TYPE) parameter, the decimal length is specified in the form (length1 length2), where length1 specifies the total number of digits in the value (including the decimal portion), and length2 specifies the number of allowable decimal digits to the right of the decimal point. The value for length2 is optional. Zero is assumed if it is not entered.
- If *CHAR, *NAME, *SNAME, *CNAME, *CMDSTR, or *VARNAME is specified for the Type of value (TYPE) parameter, only length1 is specified. It identifies the number of characters passed.
- If *HEX is specified for the Type of value (TYPE) parameter, only length1 is specified. This length specifies the number of characters passed after the hexadecimal digits have been converted to character digits. Because 2 hexadecimal digits are converted to 1 decimal digit, the number of hexadecimal digits converted is twice the value of this length.
- If *X is specified for the Type of value (TYPE) parameter, the LEN parameter is used as follows:
  - For character data, length1 specifies the minimum length to be passed. If a longer value is entered, the entire value is passed.
  - For decimal data, length2 and length3 specify the length and decimal positions for a constant value. If a variable is entered, it is passed according to the variable attributes.
  - For a logical value, length1 specifies the length of the value, which is always 1.
Return value (RTNVAL)

Specifies whether a value is returned by the command processing program through the parameter being defined in this PARM statement.

*NO  No value can be returned in the parameter being defined. The parameter is an input parameter only.

*YES  A value is to be returned by the command processing program in the parameter. A CL variable name must be specified (on the CALL command) to receive the value. *YES is valid only if *DEC, *CHAR, *LGL, *INT2, *INT4, *UINT2, *UINT4, or *X is specified for the Type of value (TYPE) parameter. Also, *YES is valid only on commands that are limited to CL programs. That is, if either *BPGM or *IPGM is specified in the Create Command (CRTCMD) command that uses the source file containing this PARM statement, *YES can be specified here. *YES must be specified on the Varying length (VARY) parameter, if *YES is specified here and on the Pass attribute byte (PASSATR) parameter. If *NO is specified for the Allow variable names (ALWVAR) parameter, or if the Maximum values allowed (MAX) parameter has a value of greater than 1, *YES is not valid. *YES is also not valid with the following parameters:

- Constant value (CONSTANT parameter)
- Default value (DFT parameter)
- Restricted values (RSTD parameter)
- Valid values (VALUES parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)
- Single values (SNGVAL parameter)
- If a file parameter, how used (FILE parameter)
- Full field required (FULL parameter)
- Value an expression (EXPR parameter)

Constant value (CONSTANT)

Specifies that a value is passed to the command processing program as a constant when the command being defined is processed; the parameter does not appear externally on the command. The value specified in this parameter (if any) must satisfy the requirements specified by the following parameters:

- Type of value (TYPE parameter)
- Value length (LEN parameter)
- Valid values (VALUES parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)
- Full field required (FULL parameter)

If a character constant is specified in this parameter, it can be no longer than 32 characters.

If a constant is specified for the parameter being defined, no prompt text can be specified for the Prompt specifications (PROMPT) parameter because the parameter will not be prompted.

This parameter is not valid for the following:

- *CMD, *NULL, or *ZEROELEM specified for the Type of value (TYPE) parameter
A value greater than 1 specified for the **Maximum values allowed (MAX)** parameter

• The **Default value (DFT)** parameter

• *YES* specified for the **Return value (RTNVAL)** parameter

• *YES* specified for the **Value an expression (EXPR)** parameter

Variables cannot be coded for this parameter.

---

**Restricted values (RSTD)**

Specifies whether the value entered for the parameter (specified in the PARM statement) is restricted to only one of the values given in the **Valid values (VALUES)** parameter, the **Special values (SPCVAL)** parameter, or the **Single values (SNGVAL)** parameter, or whether the value can be any value that satisfies the requirements specified by the following parameters:

• Type of value (TYPE parameter)

• Value length (LEN parameter)

• Relational expression (REL parameter)

• Range of values (RANGE parameter)

• Special values (SPCVAL parameter)

• Single values (SNGVAL parameter)

• Full field required (FULL parameter)

*NO*  The value entered for the parameter specified by the **Keyword (KWD)** parameter can be anything that matches the requirement specified by the following parameters in this PARM statement:

• Type of value (TYPE parameter)

• Value length (LEN parameter)

• Relational expression (REL parameter)

• Range of values (RANGE parameter)

• Special values (SPCVAL parameter)

• Single values (SNGVAL parameter)

• Full field required (FULL parameter)

*YES*  The value entered for the parameter specified by KWD in this PARM statement is restricted to one of the values in the **Valid values (VALUES)** parameter, or to one of the from-values in either the **Special values (SPCVAL)** parameter or the **Single values (SNGVAL)** parameter. *YES* cannot be specified if a statement label, *CMD, *NULL, or *ZEROELEM is specified for the **Type of value (TYPE)** parameter, or if *YES is specified for the **Return value (RTNVAL)** parameter.

---

**Default value (DFT)**

Specifies the default value that is assigned to the parameter if a value is not specified by the user. That is, the default value is used as the value of the parameter if the user omits the parameter while entering the command or if the user specifies *N as the parameter value. The default value must satisfy one of the following:

• It must match the requirements specified by the following parameters.

  – Type of value (TYPE parameter)

  – Value length (LEN parameter)

  – Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Full field required (FULL parameter)

- It must be one of the from-values in the Special values (SPCVAL) parameter, or the Single values (SNGVAL) parameter.
- If the default is a character constant, it can have no more than 32 characters.
- If *YES is specified on the Restricted values (RSTD) parameter, it must be in the list of values in the Valid values (VALUES) parameter, or in the list of from-values of either the Special values (SPCVAL) parameter or the Single values (SNGVAL) parameter.
- It must be a from-value on the Single values (SNGVAL) parameter if the parameter being defined is a list of unlike values or it is a qualified name. This is true when a statement label is specified for Type of value (TYPE) parameter; the label is used to identify a QUAL or ELEM statement.

This parameter is not valid if the Constant value (CONSTANT) parameter is specified. This parameter is valid only if 0 is specified for the Minimum values required (MIN) parameter, which means the parameter named in the Keyword (KWD) parameter is optional. No default can be specified if RTNVAL(*YES) is specified on the Return value (RTNVAL) parameter; instead, a null pointer is passed for the default. A default cannot be specified if *CMD, *ZEROELEM, or *NULL is specified on the Type of value (TYPE) parameter. If *VARNAME is specified on the Type of value (TYPE) parameter, a default special value cannot be specified; a default variable name cannot be specified.

An assumed default value is not displayed by the command prompt; a blank input field is shown instead. If a default is specified, it is displayed by the prompt exactly as specified.

value Specify the default value that meets the specified requirements or that is one of the values specified in the Valid values (VALUES) parameter, the Special values (SPCVAL) parameter, or the Single values (SNGVAL) parameter.

Variables cannot be coded for this value.

Valid values (VALUES)

Specifies a list of up to 300 constants (fixed values) from which one constant can be entered as the value of the parameter named on the Keyword (KWD) parameter. This parameter is valid only if all of the following are true:
- *YES is specified on the, Restricted values (RSTD) parameter.
- Both the Range of values (RANGE) parameter and the Relational expression (REL) parameter are not specified.
- Each constant matches the attributes specified by the following parameters.
  - Type of value (TYPE parameter)
  - Value length (LEN parameter)
  - Full field required (FULL parameter)

Character constants specified in this parameter can be no longer than 32 characters. Type the constants (not more than 300) that can be specified as the value of the parameter. This parameter is not valid if *CMD, *CMDSTR, *X, *NULL, statement label, *VARNAME, or *ZEROELEM is specified on the Type of value (TYPE) parameter, or if *YES is specified on the Return value (RTNVAL) parameter.
Relational expression (REL)

Specifies the relationship between the parameter value of this parameter and the value of a constant or another parameter. If a keyword is specified, it must be preceded by an ampersand (&) to indicate that it is the value of the keyword that is to be tested. The value associated with the referred to keyword is the value passed to the command processing program, not the user-specified value. If the relationship is with another parameter whose value is a list of values or a qualified name, the first value only is used in the comparison.

To specify the relationship, enter one of the following relational operators followed by either a constant or the keyword name of the other parameter (which must be preceded by an &).

*LT    less than
*LE    less than or equal to
*EQ    equal to
*GE    greater than or equal to
*GT    greater than
*NL    not less than
*NE    not equal to
*NG    not greater than

The REL parameter is not valid in the following cases:

• If *YES is specified on the Return value (RTNVAL) parameter.
• If either the Range of values (RANGE) parameter or the Valid values (VALUES) parameter is specified.

If a *CHAR (character type) is specified on the Type of value (TYPE) parameter, the EBCDIC value of the character string is used as an unsigned integer in the comparison. If a character constant is specified in this parameter, it can be no longer than 32 characters.

Variables can be coded for this element.

Range of values (RANGE)

Specifies the range, or limits, for the parameter value. The parameter value must be greater than or equal to the lower limit value specified, and it must be less than or equal to the upper limit value specified. For example, 15 would be valid if RANGE was specified as (0 16).

For nonnumeric data types, such as character, the range of values and the data specified are right-justified and padded on the left with blanks. A numeric range should not be used to define an interval for nonnumeric data unless leading zeros are specified or the data is only 1 character in length.

Variables can be coded for this element.

The upper and lower limits of the range can be specified either by a keyword representing the value or by the value itself. If a keyword is specified, it must be preceded by an ampersand (&) to indicate that the value of the keyword is to be tested. The value of its parameter at the time of the check is used to determine the range. The value that is tested is the value passed to the command processing program,
not the user-specified value. If the keyword identifies a list of values or a qualified name, only the first value is used as the range limit. A keyword may not refer to a parameter that is defined with *NULL specified on the Value to pass if unspecified (PASSVAL) parameter. This parameter is not valid with *NULL specified on the Value to pass if unspecified (PASSVAL) parameter.

This parameter is also not valid in the following cases:

- If *YES is specified on the Return value (RTNVAL) parameter.
- If either the Relational expression (REL) parameter or the Valid values (VALUES) parameter is specified.

Character constants specified in this parameter can be no longer than 32 characters.

Variables can be coded for this element.

### Special values (SPCVAL)

Specifies a list of up to 300 entries that define special values that can be entered on the parameter. Each entry specifies a character string (a from-value) that can be entered even though it may not meet all validity checking requirements. If the entered character string matches the from-value of one of the entries, and the to-value is specified, the string is replaced with the to-value and is then passed to the command processing program (CPP) without further checking. If the to-value is omitted, the from-value is passed to the CPP. This parameter is not valid if *YES is specified on the Return value (RTNVAL) parameter, or if *CMD, *CMDSTR, *X, *ZEROELEM, *NULL, or a statement label is specified for the Type of value (TYPE) parameter.

The from-value is a character string, but the to-value can be anything that is passable. However, for TYPE(*DATE) the to-value must be specified not quoted in the mmddyy, mmddyyyy, or the Cymmdd format. If a CL variable is used for the from-value, its type must be *CHAR. The to-value must be no longer than specified on the Value length (LEN) parameter, and, if *DEC, *INT2, *INT4, *UINT2 or *UINT4 is specified for the Type of value (TYPE) parameter, the type of the to-value must be the same. If the Type of value (TYPE) parameter is a character type (such as *CHAR, *LGL or *DATE), the to-value must be a character string. Character constants specified in this parameter can be no longer than 32 characters. If a to-value is not specified, the from-value must be passable.

If a to-value of *CURLIB is specified, the name of the current library, rather than the value *CURLIB, is passed to the CPP. If the from-value is *CURLIB and no to-value is specified, or if the to-value is *CURLIB and it is enclosed in apostrophes, the value *CURLIB is passed to the CPP.

Variables cannot be coded for this element.

### Single values (SNGVAL)

Specifies a list of up to 300 single values that can be specified for a parameter being defined as a mixed list or as a qualified name, when a statement label is specified for the Type of value (TYPE) parameter, or specifies that it is to accept two or more values as defined by the Maximum values allowed (MAX) parameter. Any one of the single values can be entered instead of a list of values or a qualified name that the parameter is defined to accept. Each entry specifies a character string (a from-value) that can be entered. If an entered character string matches the from-value of one of the entries and the to-value is
specified, the data is replaced with the to-value and is then passed to the command processing program without further checking. If the to-value is omitted, the from-value is passed to the command processing program.

The to-value (or the from-value, if the to-value is omitted) must be passable, as specified in the **Special values (SPCVAL)** parameter. Character constants specified in this parameter can be no longer than 32 characters. This parameter can be specified only if the **Maximum values allowed (MAX)** parameter is greater than 1 or if TYPE is specified as a statement label of a QUAL or ELEM statement. Each single value can only be substituted for a list of values or a qualified name; it cannot be a list item or qualifier. It is passed as the first and only element of the list.

This parameter is not valid if *YES is specified on the Return value (RTNVAL) parameter, or if *CMD, *CMDSTR, *X, *ZEROELEM, *NULL, or a statement label is specified for the Type of value (TYPE) parameter.

If a to-value of *CURLIB is specified, the name of the current library, rather than the value *CURLIB, is passed to the command processing program. If the from-value is *CURLIB and no to-value is specified, or if the to-value is *CURLIB and it is enclosed in apostrophes, the value *CURLIB is passed to the command processing program.

Variables cannot be coded for this element.

---

**Minimum values required (MIN)**

Specifies the minimum number of values that must be entered for the parameter being defined. For a parameter that does not allow multiple like values, only zero (0) for optional and 1 for required can be specified as the minimum number of values.

**Note:** Required parameter statements must precede optional statements. If required parameter statements are not specified first, the system assumes that the specified parameter is optional, and the minimum number of values for required parameters is ignored.

For a parameter that allows multiple like values, because a value greater than 1 is specified for the **Maximum values allowed (MAX)** parameter, zero (0) indicates that no values need be entered; therefore, it is an optional parameter. A value of 1 or greater than 1 indicates the minimum number of values that must be entered for the parameter, and, therefore, it is a required parameter. The value cannot exceed 1 if *NULL is specified for the Type of value (TYPE) parameter.

1. **0** The parameter is optional; it does not have to be entered.

1. **minimum-number**

   Specify the minimum number of elements that must be specified for this parameter. If 1 is the assigned value, it specifies that at least one value is required for the parameter. If a number greater than 1 is specified, the parameter is a list that must have at least as many elements as the number specified.
**Maximum values allowed (MAX)**

Specifies, if this PARM statement is defining a simple list list parameter, the maximum number of list items that this list parameter can contain. If a value greater than 1 is specified, the parameter is capable of accepting multiple like values (that is, a simple list). This support is primarily intended for IBM-supplied commands. All values entered for this parameter (at the time the command is run) must satisfy the validity checking requirements specified by the other parameter values on this PARM statement.

**Note:** The values for a list parameter are passed consecutively, preceded by a 2-byte binary value that indicates the number of values entered in the parameter by the user. CL programs do not support the handling of binary values in variables.

1 The parameter accepts only one value; the parameter is not a list parameter.

*maximum-number*

Specify the maximum number of elements that the list parameter can accept. The specified maximum must be greater than or equal to the value specified in the **Minimum values required (MIN)** parameter, and less than or equal to 300. If the maximum is greater than 1 and a statement label that identifies a QUAL or ELEM statement is not specified for the **Type of value (TYPE)** parameter, the parameter is a simple list of like elements (that is, each element in the list has the same requirements, such as type and length). If a statement label is specified and it points to the label of an ELEM or QUAL statement, a number greater than 1 should only be specified for this parameter if a list of lists or a list of qualified names is accepted. A maximum greater than 1 is not valid if *CMD, *CMDSTR, or *NULL is specified for the **Type of value (TYPE)** parameter, or if *YES is specified for the **Return value (RTNVAL)** parameter, or if the **Constant value (CONSTANT)** parameter is specified.

**Allow unprintable characters (ALWUNPRT)**

Specifies whether this PARM statement accepts the hexadecimal characters above X'FF' or those in the range of X'00' through X'3F'. This parameter is valid only if *CHAR or *X is specified for the **Type of value (TYPE)** parameter.

*YES Characters can be passed to the command processing program and sent to the display or printer.

*NO Unprintable characters cannot be passed to the command processing program.

**Allow variable names (ALWVAR)**

Specifies whether to allow variable names for the parameter. *NO is not allowed on this parameter if *VARNAME, *ZEROELEM, *NULL, or a statement label is specified on the **Type of value (TYPE)** parameter.

*YES Variable names can be used for the parameter.

*NO Variable names cannot be used for the parameter.
Is PARM a program name (PGM)

Specifies whether this parameter element is a program name. *YES is valid only if a statement label, *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC is specified for the Type of value (TYPE) parameter. Specifying *YES here does not have any effect on the parameter element being defined by the PARM statement; it only indicates to the compiler that the value for this parameter is a program name. This information is stored so that it can be included in the output of the Display Program References (DSPPGMREF) command.

*NO  The parameter defined in this PARM statement is not a program name.

*YES  The parameter defined in this PARM statement is a program name.

Is PARM a data area name (DTAARA)

Specifies whether the parameter is a data area name. *YES is valid only if a statement label, *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC is specified for the Type of value (TYPE) parameter. Specifying *YES here does not have any effect on the parameter being defined by the PARM statement; it only indicates to the compiler that the value for this parameter is a data area. This information is stored so that it can be included in the output of the Display Program References (DSPPGMREF) command.

*NO  The parameter defined in this PARM statement is not a data area name.

*YES  The parameter defined in this PARM statement is a data area name.

If a file parameter, how used (FILE)

Specifies the expected use of the file and whether the parameter is a file name. The parameter can be specified as the name of a file that has a specific use so that, at compile time, the names can be used to get file reference information about where the files are used. The specification in this parameter does not have any effect on the operation of the parameter being defined; it only indicates to the compiler that the value for this parameter is a file name and what type of file it is. This information is stored so it can be included in the output of the Display Program References (DSPPGMREF) command. This parameter is valid only if a statement label, *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC is specified for the Type of value (TYPE) parameter. It is not valid if *YES is specified on the Return value (RTNVAL) parameter.

*NO  The parameter is not a file name.

*IN   The parameter value is an input file name.

*OUT  The parameter value is an output file name.

*UPD  The parameter value is an update file name.

*INOUT  The parameter value is the name of a file that is used for both input and output.

*UNSPFD The parameter value is the name of a file, but its use cannot be specified.

The use of the file must match the type of file specified. For example, if *IN is specified, the file can be used only for input; if *UPD is specified, it can be used only to update existing records.
Full field required (FULL)

Specifies whether the number of characters in the parameter must be exactly the same as the number specified on the Value length (LEN) parameter (if specified) or its default length (if LEN is not specified).

*NO  The number of characters in the parameter can be less than that specified by the Value length (LEN) parameter.

*YES  The number of characters in the parameter must equal the number specified by LEN or the default length for that type. The exact length is valid only if *LGL, *CHAR, *NAME, *SNAME, *CNAME, *GENERIC, *VARNAME, or *HEX is specified for the Type of value (TYPE) parameter. Specifying *YES here is valid with *YES specified on the Return value (RTNVAL) parameter.

Value an expression (EXPR)

Specifies whether the parameter named in the KWD parameter can accept an expression containing a character concatenation or a built-in function (%SUBSTRING or %BIN).

Restrictions: Expressions are not allowed on parameters where *CMD, *ZEROELEM, *NULL, or a statement label is specified for the Type of value (TYPE) parameter.

*NO  The parameter value cannot be a concatenation expression or a built-in function.

*YES  The parameter value can be a concatenation expression or a built-in function.

Varying length (VARY)

Specifies whether the value that is passed to the command processing program is preceded by a length value that indicates the number of characters entered for the command parameter.

Note: The length value is the actual number of characters entered for the command parameter, with trailing blanks removed. The length value passed may be different than the defined parameter length or the declared variable length. The length of the field containing the character string data is determined by the defined length for the parameter or the declared LEN for CL program variables. The length value defines how many characters in the character string data field were actually entered for the command parameter.

Single values

*NO  The parameter value is not preceded by a length value.

Element 1: Return length value

*YES  The parameter value passed to the CPP is preceded by a field that indicates the number of characters actually specified for the parameter. *YES is valid only for the following parameter types: *CHAR, *NAME, *SNAME, *CNAME, *PNAME, *GENERIC, *LGL, *VARNAME, *CMD, *CMDSTR, and *X. *YES must be specified if PASSATR(*YES) and RTNVAL(*YES) are specified.

Element 2: Value length

*INT2  The parameter value is an integer passed as a 2-byte signed binary number.

*INT4  The parameter value is an integer passed as a 4-byte signed binary number.
Pass attribute byte (PASSATR)

Specifies whether an attribute byte is passed to the command processing program with the parameter data.

The attribute byte precedes the parameter data. If the parameter allows multiple values to be specified, an attribute byte precedes each value.

*NO  An attribute byte is not passed with the parameter.
*YES  An attribute byte is passed with the parameter.

The attribute byte has two fields:
1. The leftmost bit of the attribute byte indicates whether or not a value was specified. If the leftmost bit is '0'B, the value passed to the command processing program is a default value and was not specified in the command string. If the leftmost bit is '1'B, the value passed to the command processing program was specified in the command string.
2. The remaining seven bits describe the value passed to the command processing program when *CHAR is specified for the Type of value (TYPE) parameter.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'0000010'B</td>
<td>Meets *NAME rules, like A_B</td>
</tr>
<tr>
<td>'0000100'B</td>
<td>Meets <em>GENERIC rules, like AB</em></td>
</tr>
<tr>
<td>'1000101'B</td>
<td>Quoted character string, like 'A'</td>
</tr>
<tr>
<td>'0000101'B</td>
<td>Unquoted character string, like 5A</td>
</tr>
<tr>
<td>'1001000'B</td>
<td>Logical constant, '0' or '1'</td>
</tr>
<tr>
<td>'0001100'B</td>
<td>Hexadecimal value, like X'C1C2'</td>
</tr>
<tr>
<td>'0100001'B</td>
<td>Unsigned numeric value, like 5</td>
</tr>
<tr>
<td>'0101001'B</td>
<td>Unsigned numeric with decimal point,</td>
</tr>
<tr>
<td></td>
<td>like 5.2</td>
</tr>
<tr>
<td>'0110001'B</td>
<td>Signed numeric value, like -5</td>
</tr>
<tr>
<td>'0111001'B</td>
<td>Signed numeric with decimal point,</td>
</tr>
<tr>
<td></td>
<td>like -5.2</td>
</tr>
</tbody>
</table>

Value to pass if unspecified (PASSVAL)

Specifies whether a value is passed to the command processing program for this parameter. *NULL is not valid if the parameter is a constant parameter (a parameter in which a value has been specified for the Constant value (CONSTANT) parameter, or a parameter for which *ZEROELEM or *NULL has been specified for the Type of value (TYPE) parameter, or a list/qualified name defined by all constant ELEM or QUAL statements). *NULL also is not valid if *YES has been specified on the Return value (RTNVAL) parameter, or if the value specified for the Minimum values required (MIN) parameter is greater than zero. A DEP statement or the REL and RANGE keywords of other PARM statements may not refer to the value of a parameter defined with *NULL.

*DFT  The default value is always passed to the command processing program.
*NULL  A null pointer is passed to the command processing program if the parameter is not specified.
**Case of value (CASE)**

Specifies whether the value that is passed to the CPP is changed from lowercase to uppercase, or is preserved in the case specified on for the command parameter.

*MONO*

The parameter value is changed from lowercase to uppercase. Parameters enclosed with apostrophes preserve the case whether or not this value is specified.

*MIXED*

The parameter value is preserved in the case specified on the command parameter. This value can be specified only on *CHAR and *PNAME parameter types.

**CCSID of value (CCSID)**

Specifies the coded character set identifier (CCSID) to use when passing the parameter value.

*JOB*

If the command string was originally in Unicode, the value will be converted to the job CCSID. If the original command string was not in Unicode, the job CCSID is assumed and no conversion is done.

*UTF16*

The parameter value is converted to UTF16. If the original input was not in Unicode it is assumed to be in the job CCSID. This value can be specified only on *CHAR and *PNAME parameter types.

**List displacement (LISTDSPL)**

Specifies whether the displacement to a list within a list is 2-bytes or 4-bytes long. These displacements are generated when a parameter being passed to a CPP has a list within a list. This parameter is ignored if the value being built for the CPP does not contain a list within a list.

*INT2*

The displacement value is an integer passed as a 2-byte signed binary number.

*INT4*

The displacement value is an integer passed as a 4-byte signed binary number.

**Display input (DSPINPUT)**

Specifies whether the keyword value is shown in the job log or in a prompt display.

*YES*

The default response, *YES, indicates that the parameter value is shown on the prompt display and in the job log.

*PROMPT*

The response *PROMPT indicates that the parameter value is shown on the prompt display but not in the job log.

*NO*

The response *NO indicates that the value is not shown on either the prompt display or in the job log.
Choice text (CHOICE)

Specifies the choices text that is displayed to the right of the input field on the prompt screen. Up to 30 characters of text can be displayed.

*VALUES

The choices text is generated based on the values specified for the TYPE, RSTD, RANGE, SNGVAL, SPCVAL, and VALUES parameters. If constants are specified for the RANGE parameter, the choices text begins with the minimum value and the maximum value separated by a hyphen. If RANGE is not specified with constants as the minimum and maximum values, and RSTD(*NO) is specified, the choices text begins with a short description of the parameter type based on the value specified for the TYPE parameter. Values specified for the SNGVAL parameter are added to the choices text, in the order the values are defined in the command definition source and separated by a comma and a blank. The last entries added to the choices text are values specified for the SPCVAL or VALUES parameter, in the order the values are defined in the command definition source and separated by a comma and a blank. If there are too many values to fit in 30 characters, the last value is followed by three periods.

The following are examples of possible choices text generated by CHOICE(*VALUES):

- If TYPE(*DEC) and RANGE(1.0 999.9) and SPCVAL(*NOMAX -1)) are specified, the choices text will be:
  
- If TYPE(*NAME) and RSTD(*NO) and SNGVAL(*ALL) and SPCVAL(*LIBL *CURLIB) are specified, the choices text will be:
  
- If RSTD(*YES) and SNGVAL(*ALL) and SPCVAL(*ALRTBL *BNDDIR *CHTFMT *CLD *CLS *CMD) are specified, the choices text will be:

*NONE

No values are displayed.

*PGM A program that is called determines the values that are displayed. The program that is called is identified in Choice program (CHOICEPGM) parameter of the PARM statement.

message-identifier

Specify the message ID of the message used to retrieve the message containing the text for the possible values field. The message file specified on the Message file for prompt text (PMTFILE) parameter of the Create Command (CRTCMD) command is used to find the message.

‘choices-text’

Specify no more than 30 characters, enclosed in apostrophes.

Choice program (CHOICEPGM)

Specifies the program to be called during command prompting to fill in the possible choices text and the permissible values. This parameter must be specified if *PGM is specified on the Choice text (CHOICE) parameter and may not be specified otherwise.

Single values

*NONE

No program is identified to fill in the possible choices text and permissible values.

Qualifier 1: Choice program
name  Specifies the name of the program to be called during prompting to fill in the possible choices text or permissible values. If an exception occurs when the program is called, no possible choices text is left blank, and the list of permissible values is taken from the command.

Qualifier 2: Library

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

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

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

---

Prompt control (PMTCTL)

Specifies how prompting is to be controlled for this parameter. Prompting may be controlled by another parameter, specified by a Prompt Control (PMTCTL) statement referred to by label in this parameter, or by user request by pressing the F10 key.

*NONE  The parameter is always prompted, unless it is omitted due to selective prompting.

*PMTRQS  The parameter is not prompted unless:
  • The user requests optional parameters to be prompted.
  • A value was entered for the parameter before the prompt was called.
  • The parameter was selected by selective prompt characters.

statement label  Specify the label of the Prompt Control (PMTCTL) statement that is used to determine whether this parameter is prompted. The parameter is not prompted unless:
  • The conditions specified on the referred to PMTCTL statement have been met.
  • A value was entered for the parameter before the prompt was called.
  • The parameter was selected by selective prompt characters.

---

Prompt control program (PMTCTLPGM)

Specifies the program to be called to convert the value specified for the parameter into a value used on a Prompt Control (PMTCTL) statement. This parameter is valid only on parameters that are referred to in the Controlling keyword (CTL) parameter of a PMTCTL statement.

Single values

*NONE  No program is to be called to convert the parameter value for prompt control statements. If the parameter is specified in a Prompt Control (PMTCTL) statement, the actual value is compared in that PMTCTL statement.

Qualifier 1: Prompt control program

name  Specify the name of the program to be called to convert the parameter value.
Qualifer 2: Library

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

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

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

---

Key parameter (KEYPARM)

Specifies that this parameter is initially displayed when the command is prompted and a prompt override program was specified when the command was created or changed. If no prompt override program is specified, KEYPARM(*NO) is assumed for all parameters.

*NO  The parameter is not displayed initially.

*YES  The parameter is displayed initially.

---

Initial prompt length (INLPMTLEN)

Specifies the length of the input field initially displayed for the parameter when the command is prompted. The user can extend the field to a maximum length of 512 bytes by entering an ampersand (&) in the first position of the field, followed by a blank. INLPMTLEN is valid only if TYPE is specified as *CHAR, *NAME, *SNAME, *CNAME, *PNAME, *GENERIC, *CMDSTR, *HEX, *X, or *CMD. If FULL(*YES), RSTD(*YES), or CONSTANT are specified, INLPMTLEN(*CALC) must be specified or defaulted.

*CALC  The prompter will determine the length of the prompt field based on the type and length of the parameter.

*PWD  If the current value of system value QPWDLVL is ‘0’ or ‘1’, the prompt field will be 10 bytes long. Otherwise, the length of the prompt field will be determined by the length of the parameter. INLPMTLEN(*PWD) is valid only if TYPE is specified as *CHAR, *NAME, *SNAME, *PNAME, or *CNAME.

initial-prompt-length  Specify the initial length in bytes. Valid values are 1-12, 17, 25, 32, 50, 80, 132, 256, and 512.

---

Prompt specifications (PROMPT)

Specifies what prompt text is used for the parameter. The prompt text gives a short description of the parameter which appears next to the parameter keyword and input field when the command is prompted. Prompt text cannot be specified if *ZEROELEM or *NULL is specified for the Type of value (TYPE) parameter, or if a constant value is specified in the Constant value (CONSTANT) parameter.

Single values

*NONE  No prompt text is shown for the parameter defined by this PARM statement. This parameter is still prompted by its keyword name, but no prompt text is shown beside the keyword name.
Element 1: Prompt text or message ID

**message-identifier**
Specify the message identifier that specifies the message containing the prompt text of up to 30 characters that is shown when the parameter is prompted. If a message having the specified identifier cannot be found in the message file specified on the Message file for prompt text (PMTFILE) parameter of the Create Command (CRTCMD) command, the message identifier itself is used as the prompt text.

**'prompt-text'**
Specify the prompt text that is shown when the parameter is prompted. The text must be a character string of no more than 30 characters, enclosed in apostrophes.

Element 2: Order prompt is displayed

**relative-prompt-number**
A relative prompt number may be specified for the parameter. The relative prompt number specifies the order in which parameter keywords are prompted. This order affects only the order of prompting, not the order in which the parameters are passed to the command processing program. Parameters having prompt numbers are prompted before parameters having no prompt numbers.

---

**Examples**

**Example 1: Define a Numeric Parameter**

```
PARM  KWD(X)  TYPE(*DEC)  LEN(2)  MIN(1)  REL(*GT 5)
```

The value for the parameter named X, a 2-digit decimal number, must be entered. The value must be greater than 5.

**Example 2: Define a Parameter with Restricted Values**

```
PARM  KWD(CLASS)  TYPE(*CHAR)  LEN(1)  DFT(A) +
VALUES(A B C)  RSTD(*YES)
```

The value of the parameter named CLASS must be A, B, or C, if specified. If CLASS is not specified, the default value passed to the command processing program will be A.

**Example 3: Define a Parameter with Range of Valid Values**

```
PARM  KWD(MAXREC)  TYPE(*DEC)  LEN(3 0)  MIN(1) +
RANGE(&MINREC 500)
```

The value of the MAXREC parameter must be entered as a decimal number of 3 digits or less, with no digits to the right of the decimal point. The value must be greater than or equal to the value entered (or defaulted) for parameter MINREC and also must be less than or equal to 500.

**Example 4: Define a Simple List Parameter**

```
PARM  KWD.FILES)  TYPE(*NAME)  MIN(2)  MAX(5)
```

The FILES parameter is a homogeneous list that contains a minimum of two names and a maximum of five names.

**Example 5: Define a List Parameter with Restricted Values**
The value of the parameter named INVFNAME can be a list of up to three file names of which DEPT1, DEPT2, DEPT3, and *ALL are the valid choices. If *ALL is entered, no other values can be entered for the parameter. If this parameter is omitted, file name XXX is passed to the command processing program. If this parameter is entered through a command prompter, the prompt text for this parameter will be retrieved from message identifier USR0002 of the message file specified for the PMTFILE parameter on the Create Command (CRTCMD) command when the command is created. *ALL will be shown as the default parameter value.

Error messages

None
Program (PGM)

Where allowed to run:
- Batch program (*BPGM)
- Interactive program (*IPGM)

Threading: Yes

The Program (PGM) command is used in a CL source file to identify the start of a CL procedure that is to be compiled and to specify the parameters that are to be received by the procedure after it is compiled. If a PGM command is used, it must be the first command in the source file; if a PGM command is not used, a PGM command without parameters is assumed. The name of the CL procedure is specified on the CL command used to compile the CL source file.

The PGM command also specifies the parameters to be passed to the CL procedure, if any, when it is called for processing by another program. For information about how constants are passed, see the PARM parameter description for the CALL (CALL) command.

If the CL procedure source file is compiled to create a program (*PGM) object, the program can be called by a Call (CALL) or Transfer Control (TFRCTL) command, or by a routing entry in a subsystem description. When the program is called by a CALL or TFRCTL command, the specified parameters can be passed to it.

Parameters defined in this command must be passed when the procedure is called. The parameters passed must be of the type, length, and order specified in this command. Each of the parameter values can be a character string, a numeric value, or a CL variable. When received, each value is given a different CL variable name. Each CL variable name must be defined in the CL source file by a separate DCL (Declare) command before the procedure is compiled. Up to 255 parameters can be passed.

ILE programs and procedures will not detect parameter mismatches between the calling program or procedure and the called program or procedure. If the calling procedure passes more parameters than the called procedure expects, the called procedure will ignore the extra parameters. If the calling procedure passes fewer parameters than are specified on the called procedures PGM command, the results may be unpredictable.

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

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARM</td>
<td>Parameter CL variable names</td>
<td>Values (up to 255 repetitions): CL variable name</td>
<td>Optional, Positional 1</td>
</tr>
</tbody>
</table>
Parameter CL variable names (PARM)

Specifies one or more CL variables that are to receive the parameter values passed to this procedure. Specify a CL variable name for each of the values to be received; the name must start with an ampersand (&).

Null values, *N, cannot be specified for any parameter. The parameter values are associated with the variables in the PARM parameter in the order in which they were specified on the CALL or TFRCTL commands. The type and length of each value passed must have matching attributes in the calling and receiving programs. However, for character constants, the receiving program can specify a shorter length; when this is done, the character string passed is truncated to the length declared in the receiving program. For information on how each data type is passed, see the description of the PARM parameter in the CALL command.

Note: If a parameter value is to be changed by a CL procedure or specified as a variable on a CL command, it must be in writeable storage. For example, in C or C++, strings may be read only. If a read-only string is passed as a parameter to a CL procedure, and the CL procedure attempts to change the value of the variable or uses the variable on a CL command, the CL procedure will fail.

**CL-variable-name**

Specify the name of the CL variable to receive the value passed from the calling program. A maximum of 255 variables can be specified.

Examples

Example 1: CL Procedure Containing No Parameters

```cl
PGM
  ;
ENDPGM
```

This PGM command is the first command in a CL source file for a CL procedure that contains no parameters.

Example 2: CL Procedure Containing Two Parameters

```cl
PGM  PARM(&X &Y)
```

This is the first command in a CL source file for a CL procedure that contains two parameters, &X and &Y, that have values passed to them from the calling program or procedure.

Example 3: CL Procedure Containing Two Parameters in Positional Form

```cl
PGM  (&PARM1 &PARM2)
```

This is the first command in a CL source file for a CL procedure that specifies two parameters in positional form, &PARM1 and &PARM2. When this procedure is called, the calling program or procedure passes the parameter values to be used for &PARM1 and &PARM2.

Error messages

None
Verify TCP/IP Connection (PING)

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

The Verify TCP/IP Connection (VFYTCPCNN) command, also known as PING, tests the connectivity between a system and the remote system specified by the remote system parameter.

Notes:
- The VFYTCPCNN (PING) command cannot be used to verify IP over SNA connections.
- The local domain name is used by many applications including PING. PING appends the local domain to a host name if a domain is not specified or if a period (.) does not appear at the end of the specified host name.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMTSYS</td>
<td>Remote system</td>
<td>Character value, *INTNETADR</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>INTNETADR</td>
<td>Remote internet address</td>
<td>Character value</td>
<td>Optional</td>
</tr>
<tr>
<td>ADRVERFMT</td>
<td>Address version format</td>
<td>*CALC, *IP4, *IP6</td>
<td>Optional</td>
</tr>
<tr>
<td>MSGMODE</td>
<td>Message mode</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Response</td>
<td>*VERBOSE, *QUIET</td>
<td></td>
</tr>
<tr>
<td></td>
<td>message detail</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Summary, if</td>
<td>*COMP, *ESCAPE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>response errors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PKTLEN</td>
<td>Packet length (in bytes)</td>
<td>8-512, 256</td>
<td>Optional</td>
</tr>
<tr>
<td>NBRPKT</td>
<td>Number of packets</td>
<td>1-999, 5</td>
<td>Optional</td>
</tr>
<tr>
<td>WAITTIME</td>
<td>Wait time (in seconds)</td>
<td>1-120, 1</td>
<td>Optional</td>
</tr>
<tr>
<td>LCLINTNETA</td>
<td>Local internet address</td>
<td>Character value, *ANY</td>
<td>Optional</td>
</tr>
<tr>
<td>TOS</td>
<td>Type of service</td>
<td>*MINDELAY, *MAXTHRPUT, *MAXRLB, *MINCOST, *NORMAL</td>
<td>Optional</td>
</tr>
<tr>
<td>IPTTL</td>
<td>IP time to live (hop limit)</td>
<td>1-255, *DFT</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Remote system (RMTSYS)

Specifies the remote system name of the host with which the Verify TCP/IP operation takes place. To be successful, the name must be valid, and the remote system must be able to communicate with the local system. You can assign names to an internet address by using either of the following:
- Work with Host Table menu, which is an option on the Configure TCP/IP menu.
- Remote name server to map a remote system name to an internet address.
Host name resolution will depend on the value specified for the **Address version format (ADRVERFMT)** parameter.

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

*character-value*
Specify the remote system name to be verified.

---

**Remote internet address (INTNETADR)**
Specifies the remote internet address. Either a valid IP Version 4 or IP Version 6 address will be accepted. An IP Version 4 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.

*character-value*
Specify the internet address of the remote system. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

---

**Address version format (ADRVERFMT)**
Specifies how the host name specified for the **Remote system (RMTSYS)** parameter is to be resolved.

**C**ALC
The host name resolution method will be ‘calculated’ (determined) based on the host name entered in the RMTSYS parameter. IP Version 6 host name resolution will be performed if the system has at least one IP Version 6 address configured. If an IP Version 6 address is not found, IP Version 4 host name resolution will be performed if the system has at least one IP Version 4 address configured. The loopback address is not considered in this case as a configured address.

**IP4** Use the IP Version 4 host name resolution method.
**IP6** Use the IP Version 6 host name resolution method.

---

**Message mode (MSGMODE)**
Specifies the amount of information to be displayed.

**Element 1: Response message detail**

**V**ERBOSE
Display messages as each PING response arrives.

**Q**UIET
Display only the initial PING (VFYTCPCNN) message and the summary messages.

**Element 2: Summary, if response errors**

**C**OMP
If the PING (VFYTCPCNN) request is successful, the summary message returned is a completion message.
A monitorable escape message is returned. This is useful if you have written a program to issue the PING request and wish to monitor the PING request for errors. See the error messages section of the PING (VFYTCPNN) command help for a list of possible escape messages.

**Packet length (in bytes) (PKTLEN)**

Specifies the length (in bytes) of the packets that are sent to the remote system.

- **256** The packet length is 256 bytes.
- **8-512** Specify the number of bytes in each packet.

**Number of packets (NBRPKT)**

Specifies the number of packets that are sent to the remote system.

- **5** Five packets are sent.
- **1-999** Specify the number of packets that are sent to the remote system.

**Wait time (in seconds) (WAITTIME)**

Specifies the number of seconds to wait for the return (echo) packet before declaring this packet transfer a failure.

- **1** The system waits 1 second.
- **1-120** Specify the number of second to wait.

**Local internet address (LCLINTNETA)**

Specifies the local internet address of the interface that the outbound packets are to use. Any valid IP Version 4 or IP Version 6 address will be accepted. An IP Version 4 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.

- **ANY** Use any interface’s local internet address.

**character-value**

Specify the local internet address.

**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.
Note: This parameter is not used if IP Version 6 address resolution is used for verifying connectivity to a remote system.

*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.

---

**IP time to live (hop limit) (IPTTL)**

Specifies the IP datagram (packet) time-to-live value. The datagram is valid only for the number of router hops specified by this parameter. The time-to-live value acts as a "hop counter". The counter is decremented each time the datagram passes through a router or gateway. Limiting the validity of the datagram by the number of hops helps to prevent internet routing loops.

Note: IP Version 6 refers to this parameter as the **hop limit**.

*DFT Use the default time-to-live value.

The default time-to-live value for multicast addresses is 1. The default time-to-live value for all other addresses is specified by the IPTTL parameter of the Change TCP/IP Attributes (CHGTCPA) command.

1-255 Specify an IP datagram (packet) time-to-live value.

---

**Examples**

**Example 1: Verify TCP/IP Connection with a Specified Host Name**

```
VFYTCPNN  RMTSYS(IPHOST)  PKTLEN(100)  NBRPKT(10)
WAITTIME(15)
```

This command attempts to send 10 packets of 100 bytes each to a remote system (known to the TCP/IP configuration as IPHOST) over a TCP/IP link. Each packet transfer must take place within 15 seconds or it fails.

**Example 2: Verify TCP/IP Connection with an IP Address**

```
VFYTCPNN  RMTSYS(*INTNETADR)  INTNETADR('128.1.1.10')
PKTLEN(100)  NBRPKT(10)  WAITTIME(15)
```

This command attempts to send 10 packets of 100 bytes each to a remote system over a TCP/IP interface. The user represents the RMTSYS with its internet address 128.1.1.10, rather than with an assigned system name. Each packet transfer that takes more that 15 seconds fails.

**Example 3: Verify TCP/IP Connection with Host Name and Using a Specific Local Interface Address**
This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system over a specific TCP/IP interface that has the local address 9.2.2.3.

Because MSGMODE(*QUIET) is specified, only the primary output messages are displayed. The interface parameter is useful on multi-homed hosts to verify network connectivity through a specific physical interface.

**Example 4: Verify TCP/IP Connection with an IP Version 6 Address**

```
VFYTCPCNN  RMTSYS(*INTNETADR)
         INTNETADR('1:2:3:4:5:6:7:8')
```

This command attempts to verify the TCP/IP connection of a remote system that has the local address of 1:2:3:4:5:6:7:8.

**Example 5: Verify TCP/IP Connection with a Specified IP Version 6 Defined Host Name**

```
VFYTCPCNN  RMTSYS(IPV6HOST)
```

This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system (known to the IP Version 6 TCP/IP configuration as IPV6HOST) over a TCP/IP link.

The default "Address version format" is *CALC. Host name resolution may return multiple IP addresses for a given host name. But, in the case (*CALC), the first IP address (IP Version 4 or IP Version 6) resolved will be the address used when attempting to verify its connection over a TCP/IP link.

**Example 6: Verify TCP/IP Connection and Explicitly Use IP Version 6 Host Name Resolution**

```
VFYTCPCNN  RMTSYS(IPV6HOST)  ADRVERFMT(*IP6)
```

This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system (known to the IP Version 6 TCP/IP configuration as IPV6HOST) over a TCP/IP link.

This example differs from example 5 in that only a valid IP version 6 resolved address, for IPV6HOST, will be used when attempting to verify its connection over a TCP/IP link.

---

**Error messages**

None

*ESCAPE Messages*

**TCP3210**

Connection verification statistics: &1 of &2 successful (&3 %).

**TCP3219**

Address &1 does not match address version format &2.
PM iSeries Line Control (PMLINMON)

Where allowed to run:
• Interactive job (*INTERACT)

Threadsafe: No

Sometimes the line that PM eServer iSeries uses is in the connect pending state. This state does not allow PM eServer iSeries to access the line. The PM eServer iSeries Line Control display allows PM eServer iSeries to vary the line off, transmit the PM eServer iSeries data, and then put the line back in the connect pending state.

When you use this display you change the PM eServer iSeries transmission task (Q1PCM1) to check for line status and vary off the appropriate line (Q1PMOFF). Once the transmission is complete, the same line is placed in a connect pending state (Q1PMON).

1. Read the warning that is shown on the first display and then press Enter.
2. Use the prompt Do you want PM eServer iSeries automatic line control active? as a master control switch for the function. If you specify YES, the PM eServer iSeries function is active. If you specify NO, the function is disabled.
   If you specify NO, you do not need to define the Line Control list again when YES is specified.
   You can vary off and on a line by specifying the line only. You can vary off and on a line, controller, and device by specifying all three descriptions.
3. Verify the line, controller, and device that you defined. Press Enter to see a summary of your choices.
4. Press Enter to confirm your choices or press F12 to return to the previous display to change your entries. When you press F3, you are taken out of the PM eServer iSeries Line Control display.

Parameters

None

Examples

None

Error messages

Unknown
The Prompt Control (PMTCTL) statement specifies a condition that is tested to determine whether prompting is done for the parameters whose PARM statement referred to this PMTCTL statement. The PMTCTL statement must have a statement label that matches the label referred to in the Prompt control (PMTCTL) parameter of one or more PARM statements in the command definition source.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>Controlling keyword</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>COND</td>
<td>Controlling conditions</td>
<td>Values (up to 50 repetitions):</td>
<td>Required, Positional 2</td>
</tr>
<tr>
<td></td>
<td>operator</td>
<td>*LE, *NG, *SPCFD, *UNSPCFD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Controlling</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>keyword value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NBRTRUE</td>
<td>Number of true conditions</td>
<td>Single values: *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Other values: *Element list</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>operator</td>
<td>*LE, *NG</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Number of true</td>
<td>0-25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>conditions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LGLREL</td>
<td>Logical relation</td>
<td>*AND, *OR</td>
<td>Optional</td>
</tr>
</tbody>
</table>

### Controlling keyword (CTL)

Specifies the name of the parameter that controls the prompting. The value of the parameter specified here is compared to the value specified in the Controlling conditions (COND) parameter. If the Prompt control program prompt (PMTCTLPGM parameter) of the PARM statement is coded for the parameter specified here, the value returned by the program specified in that PMTCTLPGM parameter is compared to the values specified on the Controlling conditions (COND) parameter. If the parameter specified here is a list or qualified name, only the first list item or qualifier is compared.

### Controlling conditions (COND)

Specifies the condition against which the parameter specified on the Controlling keyword (CTL) parameter is tested. Up to 50 conditions can be specified.

*SPCFD

The condition is true, including the default value, if it is specified for the control parameter.
The condition is true only if the control parameter is not specified. It is not true if the default value is specified.

**relational-operator-value**

Specify the relational operator and value used to compare the value of the control parameter to the value specified in the **Controlling conditions (COND)** parameter. Valid values are *GT, *EQ, *NL, *LT, *NE, *LE, and *NG.

---

### Number of true conditions (NBRTRUE)

Specifies the number of conditions specified on the **Controlling conditions (COND)** parameter that must be true if the parameter is prompted for.

**ALL**  All the conditions must be true.

**relational-operator-value**

Specify the relational operator and number used to compare the number of conditions that are true to the number specified in the **Number of true conditions (NBRTRUE)** parameter. Valid values are *GT, *EQ, *GE, *NL, *LT, *NE, *LE, and *NG.

---

### Logical relation (LGLREL)

Specifies, when PMTCTL statements are in a group, the logical relationship between this PMTCTL statement and the preceding PMTCTL statements in the group. This allows conditional prompting using more than one controlling parameter.

**AND**  Performs a logical AND operation of the resulting condition for this PMTCTL statement with the previous PMTCTL statement.

**OR**  Performs a logical OR operation of the resulting condition for this PMTCTL statement with the previous PMTCTL statement.

---

### Examples

**Example 1: Selective Prompting with One Control Parameter**

A:  

```
  PMTCTL  CTL(TYPE)  COND((*EQ *)  (*EQ *LIST)) +
  NBRTRUE(*EQ 1)
```

If either TYPE(*) or TYPE(*LIST) is specified, the parameters which reference this PMTCTL statement are selected for prompting.

**Example 2: Selective Prompting with Multiple Control Parameters Using Multiple PMTCTL Statements**

B:  

```
  PMTCTL  CTL(P1)  COND((*EQ *ALL))
  PMTCTL  CTL(P1)  COND((*EQ *SOME))  LGLREL(*OR)
  PMTCTL  CTL(P2)  COND((*EQ *ALL))  LGLREL(*AND)
  PMTCTL  CTL(P1)  COND((*EQ *NONE))  LGLREL(*OR)
  PMTCTL  CTL(P2)  COND((*NE *ALL))  LGLREL(*AND)
```
The parameters which refers to this group of PMTCTL statements are selected for prompting if any of the following conditions exist:

- *ALL is specified for P1.
- *SOME is specified for P1 and *ALL is specified for P2.
- *NONE is specified for P1 and *ALL is not specified for P2.

**Error messages**

None
Position Data Base File (POSDBF)

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

The Position Database File (POSDBF) command allows you to set the position of a database file to either the beginning or end of an open file.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPNID</td>
<td>Open file identifier</td>
<td><em>Name</em></td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>POSITION</td>
<td>File position</td>
<td>*START, *END</td>
<td>Required, Positional 2</td>
</tr>
</tbody>
</table>

Open file identifier (OPNID)

Identifies the opened file to reposition. This file must be opened by either the Open Database File (OPNDBF) or Open Query File (OPNQRYF) command.

This is a required parameter.

*name* Specify the open file identifier.

File position (POSITION)

Specifies the starting or ending position of the database file.

This is a required parameter.

*START* The position of the database file is set to the start position of the member currently open. After the start position is set, a read next operation gets the first record in the member. A previous read operation gets the last record in the previous member, if *ALL* is specified for the *Overriding member (MBR)* parameter of the Override with Database File (OVRDBF) command. Otherwise, a *get past start of file* exception occurs.

*END* The position of the database file is set to the end of the member currently open. After the end position is set, a read next operation gets the first record in the next member, if *ALL* is specified for the MBR parameter of the Override with Database File (OVRDBF) command. Otherwise, a *get past end of file* exception occurs. A read previous operation gets the last record in the member.
Examples

POSDBF  OPNID(XXX)  POSITION(*START)

This command sets the record position of the database file that is opened with OPNID(XXX) to the starting position of the database file member that is currently open.

Error messages

*ESCAPE Messages

CPF5213
   Positioning of member &3 failed.

CPF5230
   No file open with OPNID(&4).
Print Adopting Objects (PRTADPOBJ)

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

The Print Adopting Objects (PRTADPOBJ) command allows you to print a report of the objects that adopt the special and private authorities of the specified user profile. This is a way to check for security exposures associated with program adoption.

Restrictions:
1. You must have *ALLOBJ or *AUDIT special authority to use this command.
2. The user profile specified on the command is locked while the command is running. The lock prevents such things as objects having their owner changed to this profile. If this profile owns a lot of objects, the profile could be locked for an extended period of time.

This command will print two reports for a user profile. The first report (Full Report) will contain all of the objects that adopt the authorities of the user profile. The second report (Changed Report) will contain the objects that now adopt the authorities of the user profile that did not adopt the authorities of the user profile when the PRTADPOBJ command was previously run for the user profile. If the PRTADPOBJ command was not previously run for the user profile, there will be no ‘Changed Report’. If the command has been previously run for the user profile but no additional objects adopt the authorities of the user profile, then the ‘Changed Report’ will be printed but there will be no objects listed.

The reports will contain the following information:
- The name of the user profile.
- The special authorities that the user profile has.
- The date and time the report was last run (shown on Changed Report only).
- An entry for each object that adopts the user profile’s authority. Each entry contains the following information:
  - The name of the object.
  - The type of object.
  - The object’s *PUBLIC authority. If the object or the object’s library is locked at the time the report is created, the value is set to *LOCKED.
  - The name of the library the object is in.
  - The library’s *PUBLIC authority. If the library is locked at the time the report is created, the value is set to *LOCKED.
  - An indication of whether there are any private authorities on the object (‘Y’ or ‘N’). If the object or the object’s library is locked at the time the report is created, the value is left blank.

Note: If there are no objects that adopt the authority of a user profile, no reports will be printed for that user. If none of the user profiles specified on the command have objects that adopt the authority of the user profiles, then there will be no reports generated.

The file QSECADPOLD in library QUSRSYS contains information from the last time the PRTADPOBJ command was run for a user profile. There is a member within the file, with the same name as the user profile, for each profile that has been previously specified on the command. System file QADPGMAD in library QSYS with format name of QSYPGMAD is the model file for the QSECADPOLD file.
### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>USRPRF</td>
<td>User profile</td>
<td>Generic name, name, *ALL</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>CHGRPTONLY</td>
<td>Changed report only</td>
<td>*NO, *YES</td>
<td>Optional, Positional 2</td>
</tr>
</tbody>
</table>

### User profile (USRPRF)

This is a required parameter.

The name of the user profile whose adopted object information will be printed.

*ALL  The adopted information will be printed for all user profiles.

name  The name of the user profile to print the adopted information for.

generic-name  The generic name of the user profile to print the adopted information for. A generic name is a character string of one or more characters followed by an asterisk (*).

### Changed report only (CHGRPTONLY)

Specifies whether just the changed report should be printed.

*NO  The full and changed reports will be printed.

*YES  Only the changed report will be printed.

### Examples

```
PRTADPOBJ USRPRF (OURSECOFR)
```

This command prints both full and changed reports for the objects that adopt the special and private authorities of the user profile OURSECOFR.

### Error messages

**ESCAPE Messages**

CPFB304  User does not have required special authorities.

CPFB307  Command &1 in use in another job.
Print AFP Data (PRTAFPDTA)

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

The Print Advanced Function Printer Data (PRTAFPDTA) command prints output received from a System/370 host. This command allows the user to specify the file being printed and the parameters used to control the print operation.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>File</td>
<td>Qualified object name</td>
<td>Required, *LIBL, *CURLIB</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: File</td>
<td>Name</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td>Positional 1</td>
</tr>
<tr>
<td>MBR</td>
<td>Member</td>
<td>Name, *FIRST</td>
<td>Optional</td>
</tr>
<tr>
<td>DEV</td>
<td>Print device</td>
<td>Name, *JOB, *SYSVAL</td>
<td>Optional</td>
</tr>
<tr>
<td>FORMDEF</td>
<td>Form definition</td>
<td>Single values: *DEVD, *INLINE</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Form definition</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td>Optional</td>
</tr>
<tr>
<td>COPIES</td>
<td>Number of copies</td>
<td>1-255, 1</td>
<td>Optional</td>
</tr>
<tr>
<td>STRPAGE</td>
<td>Starting page</td>
<td>Integer, 1</td>
<td>Optional</td>
</tr>
<tr>
<td>ENDPAGE</td>
<td>Ending page</td>
<td>Integer, *END</td>
<td>Optional</td>
</tr>
<tr>
<td>FIDELITY</td>
<td>Print fidelity</td>
<td>*ABSOLUTE, *CONTENT</td>
<td>Optional</td>
</tr>
</tbody>
</table>

File (FILE)

Specifies the Advanced Function Printing Data Stream (AFPDS) file to be printed. Only physical files are supported for this command. If you use the Override with Printer File (OVRPRTF) command with PRTAFPDTA, do not override the device type (DEVTYPE parameter).

This is a required parameter.

Qualifier 1: File

name Specify the name of the AFPDS to be printed.

Qualifier 2: Library

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

*CURLIB The current library for the job is 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.

---

**Member (MBR)**

Specifies the member that contains the data to be printed.

*FIRST*

The first member in the database file is used.

**name** Specify the name of the file member that contains the data to be printed.

---

**Print device (DEV)**

Specifies the printer that prints the file.

*JOB*

The printer device specified in the job description is used.

*SYSVAL*

The value specified in the system value QPRTDENV is used.

**name** Specify the name of the printer device.

---

**Form definition (FORMDF)**

Specifies the form definition to use when printing the file. A form definition is a resource object that defines the characteristics of the form such as: overlays, position of page data on the form, number of copies of pages, and modification to pages. The form definition is located inline with the file being printed, or in a library.

**Single values**

*DEVD*

The device description obtains the name of the form definition being used. If no value is specified, *DEVD* is assumed.

*INLINE*

The form definition that is inline with the printer file is used.

**Qualifier 1: Form definition**

**name** Specify the name of the form definition that must exist in the library named. A maximum of 8 characters can be used.

**Qualifier 2: Library**

*LIBL*

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

*CURLIB*

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

**name** Specify the name of the library to be searched.
**Number of copies (COPIES)**

Specifies, for spooled files, the number of copies being printed.

1  One copy of the output is printed.
1-255  Specify the number of copies that are to be printed.

**Starting page (STRPAGE)**

Specifies the page on which printing starts. This parameter is used for partial printing of a file.

1  Printing starts on page 1. If the start page is not specified, 1 is assumed.

integer  Specify the page number on which printing starts.

**Ending page (ENDPAGE)**

Specifies the page on which printing ends. This parameter is used for partial printing of a file ending at a specified page number. If both the start page and the end page are specified, the end page must be greater than or equal to the start page. Specifying an end page beyond the end of the actual file does not create an error condition.

*END  Printing concludes at the end of the file.

integer  Specify the page number on which printing ends.

**Print fidelity (FIDELITY)**

Specifies the degree of exactness required when printing the file.

*ABSOLUTE  The job is printed only if the file can be printed exactly as specified by the data stream and external controls.

*CONTENT  Prints the file using all available exception handling.

**Examples**

Example 1: Printing Specific Pages

PRTAFPDTA  FILE(MYLIB/MYFILE)  STRPAGE(2)  ENDPAGE(6)

This command prints the first member in file MYFILE in library MYLIB starting with page 2 and ending on page 6.
Example 2: Printing Using All Available Exception Handling

PRTAFPDTA FILE(MYLIB/MYFILE) FORMDF(F10101) FIDELITY(*CONTENT)

This command prints the first member in file MYFILE in library MYLIB using a form definition of F10101 and all available exception handling.

Error messages

*ESCAPE Messages

CPF511B
   Data stream not correct for record &2 in file &1.

PQT4001
   Data stream not valid in structured field &2 in file &1.

PQT4003
   Form definition &2 not found in library.

PQT4004
   Starting page number &1 greater than ending page number &2.

PQT4006
   Unable to process file &1 because of variable length fields.

PQT4007
   Data stream not valid in file &1.
Print Command Usage (PRTCMDUSG)

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

The Print Command Usage (PRTCMDUSG) command creates a cross-referenced listing of a specified group of CL commands that are used in a specified group of CL programs. The report shows, program by program, which of the specified commands are used in each program. The report can be used to identify which programs need to be recompiled because of changes that have been made to the command definition objects of commands specified on the PRTCMDUSG command. Note that this command can take a long time to run and can make a lot of printed output.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CMD</strong></td>
<td>Command</td>
<td>Values (up to 50 repetitions): Qualified object name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Command</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL</td>
<td></td>
</tr>
<tr>
<td><strong>PGM</strong></td>
<td>Program</td>
<td>Qualified object name</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Program</td>
<td>Generic name, name, *ALL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *USRLIBL, *CURLIB, *ALLUSR</td>
<td></td>
</tr>
</tbody>
</table>

Command (CMD)

Specifies the names of up to fifty CL commands for which specified programs are searched and printed in a report. The system searches the specified programs for every occurrence of each command you specify.

**Note:** PRTCMDUSG cannot be used to print the command usage for ILE CL programs and modules.

This is a required parameter.

Qualifier 1: Command

*name* Specify the name of a command.

Qualifier 2: Library

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

*name* Specify the name of the library that contains the CL commands whose usage in CL programs is to be reported.
Program (PGM)

Specifies one or more CL programs that are searched for the specified commands. Only the programs and libraries for which you have some (any) authority are included in the report. This parameter also can specify that all (*ALL) programs in the specified library or libraries (*USRLIBL/*ALL, for example) are searched.

Qualifier 1: Program

*ALL  All CL programs in the specified library for which the user has some authority are searched to locate the specified CL commands.

generic-name

Specify the generic name of several programs in the specified library qualifier that are searched for the specified commands. A generic name can be specified as a character string that contains one or more characters followed by an asterisk (*).

name  Specify the name of the CL program that is to be searched for the specified CL commands.

Qualifier 2: Library

*USRLIB

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

*CURLIB

The current library for the job is used to search for the CL program. If no library is specified as the current library for the job, QGPL is used.

*ALLUSR

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

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

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

QDSNX  QRLxxxx  QUSRRIJS  QUSRVRxRxMx
QGPL  QSRVAGT  QUSRINFSKR
QGPL38  QSYS2  QUSRNOTES
QMTC  QSYS2xxx  QUSROND
QMGTC2  QIS36F  QUSRPOS65
QMPGDATA  QUSER3B  QUSRPOSSA
QMQMDATA  QUSRADSM  QUSRPMVVR
QMOPROC  QUSRBRM  QUSRDRDS
QPFRDATA  QUSRDRICL  QUSRYS
QRCR  QUSRDIRDB  QUSRVI

1. ‘xxxxx’ is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVRxRxMx, 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 QUSRVRxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

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

PRTCMDUSG  CMD(CPYF)  PGM(PAYROLL/*ALL)

This command searches all CL programs in the library PAYROLL for the Copy File (CPYF) commands and prints the names of both the commands and the program.

Error messages

*ESCAPE Messages

CPF0593
PRTCMDUSG command ended by controlled end.

CPF0595
PRTCMDUSG command ended.

CPF0596
PRTCMDUSG command ended. Cannot open print file.
Print Communications Security (PRTC MNSEC)

Where allowed to run: All environments (*ALL)

Threadsafe: No

The Print Communications Security (PRTC MNSEC) command allows you to print a report containing the security attributes of the *DEVD, *CTLD and *LIND objects currently on the system. This command provides a way to check the security of your communications configuration on the system.

The Print Communications Security command will create two spooled output files containing communications security information. The first spooled output file will contain a report generated by the Display Configuration List (DSPCFGL) CL command. This report will contain the entries currently in the APPN remote configuration list QAPPNRMT. If the QAPPNRMT configuration list does not exist on the system then no report will be printed. The second spooled output file contains the security attributes of the *DEVD, *CTLD and *LIND objects on the system.

Restriction: You must have *ALLOBJ and *IOSYSCFG, or *AUDIT special authority to use this command.

The spooled output file containing the *DEVD, *CTLD and *LIND objects will contain two reports. The first report (Full Report) will contain all of the communications objects and will print the security attributes of each object. The second report (Changed Report) will contain the communications objects that have changed since the PRTC MNSEC command was last run. If the PRTC MNSEC command was not previously run, there will be no 'Changed Report'. If the command has been previously run but no communication object information has changed then the 'Changed Report' will be printed but there will be no objects listed.

The first report will contain the entries from the APPN remote configuration list object QAPPNRMT. If the QAPPNRMT configuration list does not exist then no report will be printed.

The second report will contain the information listed below. The report lists *DEVD, *CTLD and *LIND object types. Some fields may be blank or set to zero if the field does not apply to the type of object listed in the report.

*DEVD object types:

The reports will contain the following information:

- The object type being reported.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each *DEVD object on the system.
  - The name of the communications object.
  - The object type of the communications object.
  - The device category of the communications object.
  - The secure location value of the communications object.
  - An indication if there is a location password for the communications object.
  - The APPN capable value of the communications object.
  - The single session value of the communications object.
  - The pre-establish session value of the communications object.
  - The SNUF program start value of the communications object.
*CTLD object types:

The reports will contain the following information:

- The object type being reported.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each *CTLD object on the system.
  - The name of the communications object.
  - The object type of the communications object.
  - The controller category of the communications object.
  - The auto create value of the communications object.
  - The switched controller value of the communications object.
  - The call direction value of the communications object.
  - The APPN capable value of the communications object.
  - The CP sessions value of the communications object.
  - The disconnect timer value of the communications object.
  - The auto delete minutes value of the communications object.
  - The device name value of the communications object.

*LIND object types:

The reports will contain the following information:

- The object type being reported.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each *LIND object on the system.
  - The name of the communications object.
  - The object type of the communications object.
  - The line category of the communications object.
  - The auto create value of the communications object.
  - The auto delete minutes value of the communications object.
  - The auto answer value of the communications object.
  - The auto dial value of the communications object.

The file QSECCMNOLD in library QUSRSYS contains information from the last time the PRTCMNSEC command was run for a library. System file QASECCMN in library QSYS with format name of QSECCMN is the model file for the QSECCMNOLD file.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHGRPTONLY</td>
<td>Changed report only</td>
<td>*NO, *YES</td>
<td>Optional, Positional 1</td>
</tr>
</tbody>
</table>
Changed report only (CHGRPTONLY)

Specifies whether just the changed report should be printed.

*NO  The full and changed reports will be printed.
*YES Only the changed report will be printed.

Examples

PRTCMNSEC

This command prints both full and change report for the communication security information.

Error messages

*ESCAPE Messages

CPFB307
Command &1 in use in another job.
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPHT (Merge TCP/IP Host Table)
Print Communications Trace (PRTCMNTRC)

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

The Print Communications Trace (PRTCMNTRC) command transfers the communications trace data for the specified line, network interface description, or network server description to a spooled file or an output file.

The PRTCMNTRC command can also be used to format communications trace data that was previously dumped to a stream file using the Dump Communications Trace (DMPCMNTRC) command.

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 (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
• The trace data for network server description traces can only be transferred to a spooled file. The trace data cannot be transferred to an output file. There are no formatting options available.
• The following user profiles have authority to this command:
  – QSECOFR
  – QSRV

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFGOBJ</td>
<td>Configuration object</td>
<td>Name</td>
<td>Optional, Positional 1</td>
</tr>
<tr>
<td>FROMSTMF</td>
<td>From stream file</td>
<td>Path name</td>
<td>Optional</td>
</tr>
<tr>
<td>CFGTYPE</td>
<td>Type</td>
<td>*LIN, *NWI, *NWS</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Output</td>
<td>*PRINT, *OUTFILE</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTFILE</td>
<td>File to receive output</td>
<td>Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: File to receive output</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>OUTMBR</td>
<td>Output member options</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Member to receive output</td>
<td>Name, *FIRST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Replace or add records</td>
<td>*REPLACE, *ADD</td>
<td></td>
</tr>
<tr>
<td>CODE</td>
<td>Character code</td>
<td>*EBCDIC, *ASCII, *CALC</td>
<td>Optional</td>
</tr>
<tr>
<td>SLTLIND</td>
<td>Line description</td>
<td>Name, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>SLCTLD</td>
<td>Controller description</td>
<td>Name, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTSNA</td>
<td>Format SNA data only</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTRR</td>
<td>Format RR, RNR commands</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------</td>
<td>---------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>FMTTCP</td>
<td>Format TCP/IP data</td>
<td>*LINTYPE, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTLCP</td>
<td>Format LCP data</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTNCNP</td>
<td>Format NCP data</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>TCPIPADR</td>
<td>Format TCP/IP data by address</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Element 1: Source/destination IP address</td>
<td>Character value, *ALL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Element 2: Source/destination IP address</td>
<td>Character value, *ALL</td>
</tr>
<tr>
<td>SLTPORT</td>
<td>IP port number</td>
<td>Decimal number, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTUI</td>
<td>Format UI data only</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTMAC</td>
<td>Format MAC or SMT data only</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTETH</td>
<td>Format Ethernet data only</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTCCD</td>
<td>Format call control data</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTBCD</td>
<td>Format broadcast data</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>EXCLMI</td>
<td>Exclude LMI data</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTLMI</td>
<td>Format LMI data only</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTHPRIP</td>
<td>Format HPR over IP data only</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>FMTLDLCIP</td>
<td>Format LDLC over IP data only</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**Configuration object (CFGOBJ)**

Specifies the configuration object being traced. The object must be a line description, a network interface description, or a network server description.

Either the CFGOBJ and CFGTYPE parameters or the FROMSTMF parameter must be specified.

*name* Specify the name of the configuration description object.

**From stream file (FROMSTMF)**

Specifies the path name of the stream file from which communications trace data is formatted. This file must have been created by running the Dump Communications Trace (DMPCMNTRC) CL command. Either the CFGOBJ and CFGTYPE parameters or the FROMSTMF parameter must be specified.

*path-name* Specify the path name of the stream file created by the DMPCMNTRC command.
Type (CFGTYPE)

Specifies the type of configuration description that was traced.

Either the CFGOBJ and CFGTYPE parameters or the FROMSTMF parameter must be specified.

*LIN  The type of configuration object is a line description.
*NWI  The type of configuration object is a network interface description.
*NWS  The type of configuration object is a network server description.

Output (OUTPUT)

Specifies whether the output from the command is printed with the job’s spooled output or sent to a database file.

Note: For network server description traces, *PRINT must be specified for this parameter.

*PRINT  The output is printed with the job’s spooled output.
*OUTFILE  The output is directed to the database file specified for the File to receive output (OUTFILE) parameter.

File to receive output (OUTFILE)

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

Qualifier 1: File to receive output

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

Qualifier 2: Library

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

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

Character code (CODE)

Specifies whether the extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII) character code is used on the line.

*CALC

The system determines whether to format the user data in EBCDIC or ASCII, based on the type of controller that is used.

*ASCII The ASCII character code is used.

*EBCDIC The EBCDIC character code is used.

Line description (SLTLIND)

Specifies whether to format data for all lines or a specific line communicating on the network during a trace.

*ALL Formats the data for all lines.

name Specify the name of the line for which trace data is formatted.

Controller description (SLTCTLD)

Specifies whether to format data for all controllers or a specific controller communicating on the network during a trace.
*ALL  Formats data for all controllers.

controller-name
   Specify the name of the controller for which trace data is formatted.

Format SNA data only (FMTSNA)
Specifies whether line protocol data or Systems Network Architecture (SNA) data is formatted. Line protocol data includes SDLC, X.25, Carrier Sense Multiple Access with Collision Detection (CSMA/CD), Ethernet DIX V2, DDI, wireless, and IBM Token-Ring Network (TRLAN).

*NO    Only line protocol data is formatted.
*YES   Only SNA data is formatted.

Format RR, RNR commands (FMTRR)
Specifies whether receiver ready (RR) and receiver not ready (RNR) commands are formatted with other data.

*NO     RR and RNR commands are not formatted with other data.
*YES    RR and RNR commands are formatted with other data.

Format TCP/IP data (FMTTCP)
Specifies whether line protocol data or Transmission Control Protocol/Internet Protocol (TCP/IP) data is formatted.

Note: If the trace data is being formatted from a stream file (FROMSTMF parameter), this parameter is ignored and TCP/IP data is formatted.

*LINTYPE
   For X.25, Ethernet, DDI, wireless, Token-Ring, and Frame Relay lines, only line protocol data is formatted. For all other lines supporting TCP/IP, TCP/IP data is formatted.
*YES   TCP/IP data is formatted.
*NO    TCP/IP data is not formatted.

Format LCP data (FMTLCP)
Specifies whether Link Control Protocol (LCP) data is included in the formatted communications trace.

Note: If FMTLCP, FMTNCP, and FMTTCP are all specified *NO when formatting data for a Point-to-Point Protocol (PPP) line, then asynchronous and unrecognized data will be placed in the spooled file. This is also the case if all are specified *YES (or *LINTYPE for FMTTCP). In all other cases asynchronous and unrecognized data will be omitted.

*YES   LCP data is formatted.
Format NCP data (FMTNCP)

Specifies whether Network Control Protocol (NCP) data is included in the formatted communications trace.

*YES  NCP data is formatted.
*NO   NCP data is not formatted.

Format TCP/IP data by address (TCPIPADR)

Specifies an internet address pair for which TCP/IP data is formatted. Any values that are valid for IP address 1 are also valid for IP address 2.

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.

For IPv6 (IP version 6) addresses, the form is x:xxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx, where x is any valid hexadecimal digit 0 through F.

Note: IPv6 addresses are only valid when formatting trace data from a stream file.

Element 1: Source/destination IP address

*ALL  The communications between the systems specified for element 2 and all other systems are printed.
character-value  Specify the address of the system for which communications between this system and the systems specified for element 2 are printed.

Element 2: Source/destination IP address

*ALL  The communications between the systems specified for element 1 and all other systems are printed.
character-value  Specify the address of the system for which communications between this system and the systems specified for element 1 are printed.

IP port number (SLTPORT)

Specifies whether data for all internet protocol (IP) ports or only a single IP port is formatted.

Note: This parameter is valid only if FMTTCP(*YES) is specified.

*ALL  Data for all IP ports is formatted.
Specify the IP port number (1 to 65535) whose data is to be formatted.

**Format UI data only (FMTUI)**

Specifies whether line protocol data or unnumbered information (UI) data is formatted.

- **NO** All line protocol data is formatted.
- **YES** Only UI data is formatted.

**Format MAC or SMT data only (FMTMAC)**

Specifies whether line protocol data or medium access control (MAC) or Station Management (SMT) data is formatted.

- **NO** The line protocol data (TRLAN or Ethernet) is formatted.
- **YES** Only MAC or SMT data is formatted.

**Format Ethernet data only (FMTETH)**

Specifies whether IEEE 802.3 data or Ethernet V2 data is formatted.

- **YES** Both IEEE 802.3 data and Ethernet V2 data are formatted.
- **NO** Only IEEE 802.3 data is formatted.

**Format call control data (FMTCCD)**

Specifies whether all network interface data or only Integrated Services Digital Network (ISDN) signalling data is formatted.

- **NO** All network interface data is formatted.
- **YES** Only ISDN signaling data is formatted.

**Format broadcast data (FMTBCD)**

Specifies whether broadcast data and data received containing destination MAC addresses is formatted.

- **YES** Broadcast data is formatted.
- **NO** Broadcast data is not formatted.
Exclude LMI data (EXCLMI)
Specifies whether to exclude local management interface (LMI) data from the formatted output.
*NO LMI data is not excluded from the formatted output.
*YES LMI data is excluded from the formatted output.

Note: You cannot specify *YES for both the EXCLMI and FMTLMI parameters.

Format LMI data only (FMTLMI)
Specifies whether local management interface (LMI) data is formatted.
*NO LMI data is not formatted.
*YES LMI data is formatted.

Note: You cannot specify *YES for both the EXCLMI and FMTLMI parameters.

Format HPR over IP data only (FMTHPRIP)
Specifies whether High Performance Routing Protocol (HPR) over IP data is included in the formatted communications trace.
*NO HPR over IP data is not formatted.
*YES Only HPR over IP data is formatted.

Format LDLC over IP data only (FMTLDLCIP)
Specifies whether Logical Data Link Control (LDLC) over IP data is included in the formatted communications trace.
*NO LDLC over IP data is not formatted.
*YES Only LDLC over IP data is formatted.

Examples
PRTCMNTRC CFGOBJ(*QESLINE) CFGTYPE(*LIN)
This command prints communications trace data for line description QESLINE.

Error messages
*ESCAPE Messages
CPF2634
Not authorized to object &1.

CPF39AF
Trace is ending - please wait

CPF39A7
Trace storage not available in communications processor

CPF39A8
Not authorized to communications trace service tool

CPF39A9
Error occurred during communications trace function

CPF39BA
Formatting options selected not valid

CPF39BB
Communications trace data not printed

CPF39BC
Communications trace print request cannot be completed

CPF39B0
No communications traces exist.

CPF39B1
Trace &1 type &2 does not exist

CPF39B3
Trace &1 type &2 contains no data

CPF39B4
Trace data for &1 type &2 cannot be printed

CPF39B5
Communications trace data not printed

CPF39B6
Communications trace function cannot be performed

CPF39B7
Trace data for &1 type &2 cannot be printed

CPF39B8
No SNA data found in trace &1 type &2

CPF39B9
No trace records found for printing trace &1 type &2

CPF39C4
IP address not valid.

CPF3CF2
Error(s) occurred during running of &1 API.

CPF9803
Cannot allocate object &2 in library &3.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.
CPF9847
   Error occurred while closing file &1 in library &2.

CPF9860
   Error occurred during output file processing.

CPF9872
   Program or service program &1 in library &2 ended. Reason code &3.

CPF98A2
   Not authorized to &1 command.

CPFA0D4
   File system error occurred. Error number &1.
Print Device Addresses (PRTDEVADR)

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

The Print Device Addresses (PRTDEVADR) command provides a printed list of addresses and related information for devices attached to a local or remote work station controller. For each device attached to the local work station controller named in the controller description (CTLD parameter), the output shows the device’s name, its port and switch setting, its type and model number, its shared session number (valid only if device type is 3486 or 3487), and whether the device is a display station or printer.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTLD</td>
<td>Controller description</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
</tbody>
</table>

Controller description (CTLD)

Specifies the name of the local or remote work station controller for which device address information is printed.

This is a required parameter.

Examples

PRTDEVADR CTLD(CTL01)

This command prints device address information for the devices that are attached to the CTL01 work station controller.

Error messages

*ESCAPE Messages

CPF2602
Controller &1 not found.

CPF2625
Not able to allocate object &1.

CPF2628
Device description previously deleted.
CPF263B
Controller &1 not a work station controller.

CPF2634
Not authorized to object &1.

CPF2778
Controller description &1 damaged.

CPF9846
Error while processing file &1 in library &2.

CPF9850
Override of printer file &1 not allowed.
The Print Directory Information (PRTDIRINF) command is used to print directory information for objects in the Integrated File System that was collected by the Retrieve Directory Information (RTVDIRINF) command. A spooled file with file name QPEZDIR goes to the spool queue associated with the job using this command.

**Restrictions:**
- You must have all object (*ALLOBJ) special authority to run this command.
- Job CCSID value 65535 is not allowed.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RPTTYPE</strong></td>
<td>Type of report</td>
<td>*DIR, *OBJ, *OWN</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td><strong>INFFILEPFX</strong></td>
<td>Information file prefix</td>
<td>Simple name, *LAST</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>INFLIB</strong></td>
<td>Information library</td>
<td>Name, *LAST</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>OWNER</strong></td>
<td>Owners</td>
<td>Generic name, name, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>OBJ</strong></td>
<td>Objects</td>
<td>Path name, *ALL, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>SUBTREE</strong></td>
<td>Directory subtree</td>
<td>*ALL, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>MINSIZE</strong></td>
<td>Smallest size</td>
<td>0-99999, 0</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>SORT</strong></td>
<td>Sort by</td>
<td>*SIZE, *OWNER, *LSTCHG, *NAME</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>INCPATH</strong></td>
<td>Include path names</td>
<td>*ALL, *NONE</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Type of report (RPTTYPE)

Specifies the type of report to print.

This is a required parameter.

*DIR  A report of space being used by each directory is printed. Information will include a total size for the outermost directory and all nested subdirectories. Information for each subdirectory will include the total of space for just the subdirectory and will not include space for objects in any directories contained in the subdirectory.

*OBJ  A report of object information for files and directories is printed.

*OWN  A report of the user profile (owner) information for files and directories is printed.

Information file prefix (INFFILEPFX)

Specifies the file name prefix of the database files that were created by the Retrieve Directory Information (RTVDIRINF) command to store the retrieved directory information.

*LAST  The database files created by the most recent invocation of the RTVDIRINF command will be used. Informational message CPI1E31 will be sent to the job log and will contain the name and library of the files used.

information-file-prefix

Specify the same file prefix as was specified on a previous invocation of the RTVDIRINF command. The RTVDIRINF command created multiple database files to store the retrieved directory information. If a file prefix is specified, a value other than *LAST must be specified for the INFLIB parameter.

Information library (INFLIB)

Specifies the library that contains the database files that were created by the Retrieve Directory Information (RTVDIRINF) command to store the retrieved directory information.

*LAST  The library used by the most recent invocation of the RTVDIRINF command will be used to find the database files which contain the retrieved directory information. Informational message CPI1E31 will be sent to the job log and will contain the name and library of the files used.

library-name

Specify the name of the library that contains the database files created by a previous invocation of the RTVDIRINF command. If a library name is specified, a value other than *LAST must be specified for the INFFILEPFX parameter.

Owners (OWNER)

Specify the names of the owners (user profiles) of the objects to print information about.

*ALL  The report contains information about objects owned by any user profile.
owner-name

Specify the user profile that owns the objects to print information about.

generic-name

Specify the generic user profile that owns the objects to print information about. 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 user profiles with names that begin with the generic prefix.

Objects (OBJ)

Specifies the names of the objects to print information about.

*ALL

If you specify an owner (OWNER parameter), all objects owned by the specified owner are included. If OWNER(*ALL) is specified, all objects are included in the report.

*NONE

No detail object information is included in the report, just a total size of owned objects, if *NONE is specified for Object (OBJ) parameter and *OWN is specified for the Type of report (RPTTYPE) parameter.

object-name

The object information is included only for the objects specified by the given name. If an owner is specified (OWNER parameter), only the objects meeting the owner criteria and that match the given name are included.

generic-name

The object information is included only for the objects that match the specified 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. If an owner is specified (OWNER parameter), only the objects meeting the owner criteria and that match the given generic name are included.

Object types (OBJTYPE)

Specifies the object types to print information about. If the OWNER or OBJ parameters were specified with a value other than *ALL, the owner and object name criteria must also be met.

Single values

*ALL

Information about objects of all types is printed.

Other values (up to 60 repetitions)

object-type

Information about objects of the type(s) specified is printed.

Directory subtree (SUBTREE)

Whether to print subdirectories statistics in the report or not.

Print Directory Information (PRTDIRINF) 227
Statistics are included for all subdirectories within the directory processed by the Retrieve Directory Information (RTVDIRINF) command.

Statistics are included only for objects in the directory processed by the Retrieve Directory Information (RTVDIRINF) command.

### Smallest size (MINSIZE)

Specifies the size of the smallest object to include.

- **0**: All objects are included regardless of size.
- **size**: Specify size in number of kilobytes.

### Sort by (SORT)

Specifies the order in which the information should be sorted.

- **SIZE**: Information is sorted by object size, from largest to smallest.
- **OWNER**: The information is sorted in alphabetical order by owner name.
- **LSTCHG**: The information is sorted by last-change date with the oldest information first.
- **NAME**: Information is sorted by object name, names are listed in alphabetical order.

### Include path names (INCPATH)

Specifies if the report will include the path where the objects reside. If *NONE is specified for Object (OBJ) parameter and *OWN is specified for the Type of report (RPTTYPE) parameter, this parameter will be ignored.

- **ALL**: Paths are included in the report.
- **NONE**: Paths are not included in the report.

### Examples

**Example 1: Print Information, Grouped by Owner, for Most Recent RTVDIRINF**

```
PRTDIRINF  RPTTYPE(*OWN)  INFFILEPFX(*LAST)  INFLIB(*LAST)
           OWNER(*ALL)  OBJ(*ALL)  OBJTYPE(*ALL)  SORT(*SIZE)
```
This command prints an owner report from the database file created by the most recent invocation of the Retrieve Directory Information (RTVDIRINF) command. Information in the report will be included for all objects, grouped by their owner. The information is sorted by object size and sent to the printer file QPEZDIR.

**Example 2: Print a Directory Report**

PRTDIRINF  RPTTYPE(*DIR)  INFILEPFX(MYROOTDIR)
            INFLIB(QUSRSYS)  SUBTREE(*ALL)

This command prints a directory report from database file MYROOTDIRO and MYROOTDIRD created by a prior invocation of the Retrieve Directory Information (RTVDIRINF) command. If information was collected for subdirectories by the RTVDIRINF command, that information will be included in the directory report. The information is sent to the printer file QPEZDIR.

---

**Error messages**

*ESCAPE Messages*

CPF2110
Library &1 not found.

CPF1ED2
File &1 is in use and cannot be accessed.

CPF1ED5
File prefix &1 or library &2 not found in QAEZDBFILE.

CPF1ED6
File &1 in library &2 not found.

CPF1ED9
Retrieved directory information not complete.

CPF1EEC
Not authorized to file &1.

CPF1E99
Unexpected error occurred.
Print Document (PRTDOC)

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

The Print Document (PRTDOC) command permits the user to print a document using the word processing function of OfficeVision.

This command also permits the user to override all print option values that are currently stored with a document. When a document is created, a set of default print options is associated with that document. If the user wants to override one or more of the parameters in this print command, the user must select OPTIONS(*YES) so that the print options appear on the display. When the print options appear, any of the print parameters can be changed. The user can override one or all of the print option parameters with this command.

Restriction: To use this command, you must be signed on as QPGMR, QSYSOPR, QSRV, or QSRVBAS, or have *ALLOBJ authority.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOC</td>
<td>Document</td>
<td>Character value, *PRV, *ALL</td>
<td>Optional, Key, Positional 1</td>
</tr>
<tr>
<td>FLR</td>
<td>Folder</td>
<td>Character value, *PRV</td>
<td>Optional, Key, Positional 2</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>Display print options</td>
<td>*NO, *YES, *PRTFILE, *OUTFILE</td>
<td>Optional, Key</td>
</tr>
<tr>
<td>PRTFILE</td>
<td>Printer file</td>
<td>Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Printer file</td>
<td>Name, QSYSPT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>OUTFILE</td>
<td>File to receive output</td>
<td>Single values: *PRV Other values: Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: File to receive output</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>OUTMBR</td>
<td>Output member options</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Member to receive output</td>
<td>Name, *FIRST, *PRV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Replace or add records</td>
<td>*REPLACE, *ADD</td>
<td></td>
</tr>
<tr>
<td>CURSTS</td>
<td>Current status value</td>
<td>Character value, *PRV, *NOCHK</td>
<td>Optional</td>
</tr>
<tr>
<td>NEWSTS</td>
<td>New status value</td>
<td>Character value, *PRV, *NOCHG</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTDTATYP</td>
<td>Type of data for output</td>
<td>*PRV, *ALL, *IDP</td>
<td>Optional</td>
</tr>
<tr>
<td>DLTDOC</td>
<td>Delete document</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Output device</td>
<td>*SAME, *PRINT, *</td>
<td>Optional</td>
</tr>
<tr>
<td>DEV</td>
<td>Print device</td>
<td>Name, *SAME, USRPRF, *SYSVAL, *WRKSTN</td>
<td>Optional</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>OUTQ</td>
<td>Output queue</td>
<td>Single values: *SAME, *FILE, *DEV, *WRKSTN Other values: Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Output queue</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL</td>
<td></td>
</tr>
<tr>
<td>SPLFILE</td>
<td>Output file</td>
<td>Name, *SAME, *DOC, *FILE</td>
<td>Optional</td>
</tr>
<tr>
<td>FORMTYPE</td>
<td>Form type</td>
<td>Character value, *SAME, *STD</td>
<td>Optional</td>
</tr>
<tr>
<td>COVERPAGE</td>
<td>Print separator page</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>PRTQTY</td>
<td>Print quality</td>
<td>*SAME, *LETTER, *TEXT, *DRAFT</td>
<td>Optional</td>
</tr>
<tr>
<td>COPIES</td>
<td>Number of copies</td>
<td>1-99, *SAME</td>
<td>Optional</td>
</tr>
<tr>
<td>DUPLEX</td>
<td>Print on both sides</td>
<td>*SAME, *YES, *NO, *TUMBLE</td>
<td>Optional</td>
</tr>
<tr>
<td>AUTOBIND</td>
<td>Automatic page binding</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>HOLD</td>
<td>Delay printing</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>PRTRRLOG</td>
<td>Print document error log</td>
<td>*PRV, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>ERRFORM</td>
<td>Error log form type</td>
<td>Character value, *SAME, *STD</td>
<td>Optional</td>
</tr>
<tr>
<td>LARGEPRINT</td>
<td>Large print</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>QRYDFN</td>
<td>Query</td>
<td>Single values: *SAME Other values: Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Query</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL</td>
<td></td>
</tr>
<tr>
<td>DTADOC</td>
<td>Data document</td>
<td>Character value, *SAME</td>
<td>Optional</td>
</tr>
<tr>
<td>DTAFLR</td>
<td>Data folder</td>
<td>Character value, *SAME</td>
<td>Optional</td>
</tr>
<tr>
<td>DTAFILE</td>
<td>Data file</td>
<td>Single values: *SAME Other values: Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Data file</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>DTAMBR</td>
<td>Data member</td>
<td>Name, *SAME, *FIRST, *FILE, *LAST</td>
<td>Optional</td>
</tr>
<tr>
<td>MLTLINRPT</td>
<td>Multiple line report</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>ADJLINES</td>
<td>Adjust line endings</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>ADJPGN</td>
<td>Adjust page endings</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>ALWWIDOW</td>
<td>Allow widow lines</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>RENUMBER</td>
<td>Renumber system page numbers</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>PRTCHGSYM</td>
<td>Print change symbols</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>SYMBOLS</td>
<td>Change symbols to print</td>
<td>Character value, *SAME</td>
<td>Optional</td>
</tr>
<tr>
<td>DRAFTSPACE</td>
<td>Draft spacing</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>LINNBR</td>
<td>Print line numbers</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>RESOLVE</td>
<td>Resolve instructions</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>LEFTSPACES</td>
<td>Additional spaces to left</td>
<td>0-99, *SAME</td>
<td>Optional</td>
</tr>
<tr>
<td>CHRID</td>
<td>Character identifier</td>
<td>*SAME, *BLANK</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Graphic character set</td>
<td>1-9999, *SAME, *BLANK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Code page</td>
<td>1-9999</td>
<td></td>
</tr>
<tr>
<td>SAVOUTPUT</td>
<td>Save resolved output</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>SAVDOC</td>
<td>Resolved output document</td>
<td>Character value, *SAME, *BLANK</td>
<td>Optional</td>
</tr>
<tr>
<td>SAVFLR</td>
<td>Resolved output folder</td>
<td>Character value, *SAME, *BLANK</td>
<td>Optional</td>
</tr>
<tr>
<td>JOBQ</td>
<td>Place on job queue</td>
<td>*SAME, *YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>---------</td>
<td>-------</td>
</tr>
</tbody>
</table>
| JOBD    | Job description | Single values: *SAME  
Other values: Qualified object name | Optional |
|         | Qualifier 1: Job description | Name | |
|         | Qualifier 2: Library | Name, *LIBL | |
| SNDMSG  | Send completion message | *SAME, *YES, *NO | Optional |
| CNLERR  | Cancel on error | *SAME, *YES, *NO | Optional |
| STRPAGE | Start page | 0.01-9999.99, *FIRST, *LAST, *PAGERANGE, *SAME | Optional |
| PAGERANGE | Page ranges | Single values: *SAME  
Other values (up to 7 repetitions): Element list | Optional |
|         | Element 1: Start page | 0.01-9999.99, *FIRST, *LAST | |
|         | Element 2: End page | 0.01-9999.99, *FIRST, *LAST, *STRPAGE | |
| LBLACROSS | Number of labels across page | 1-99, *SAME | Optional |
| LBLWIDTH | Width of labels | 2-198, *SAME | Optional |
| SHEETFEED | Sheet feed labels | *SAME, *YES, *NO | Optional |
| LBLDOWN  | Number of rows per sheet | 1-99, *SAME | Optional |
| SHFLEFTMAR | Shift left margin | *SAME, *YES, *NO | Optional |

**Document (DOC)**

Specifies the name of the document that is printed.

*PRV  The name used in the previous session is used.

*ALL  All documents to which the user is authorized are printed to a database file. This is valid only when the output is directed to an OUTFILE.

document-name  
Specify the name of the document to be printed.

**Folder (FLR)**

Specifies the name of the folder that contains the document that is printed.

*PRV  The name used in the previous session is used.

folder-name  
Specify the name of the folder that contains the document being printed.

**Display print options (OPTIONS)**

Specifies whether the print options for this document are displayed before the document is printed.
The print options are not displayed before the document is printed.

*YES  The print options are displayed before the document is printed.

*PRTFILE
The print options specified on the Printer file prompt (PRTFILE parameter) are used.

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

Printer file (PRTFILE)
Specifies the printer file to use for the print options. This parameter is valid only if *PRTFILE is also specified on the Display print options prompt (OPTIONS parameter).

When *PRTFILE is specified, the following Print Options are overridden by the appropriate values in the printer file:
• DEV (Printer ID)
• PRTQLTY
• OUTQ
• FORM
• COPIES
• HOLD
• DUPLLEX
• OUTPUT FILE

QSYSPRT
The document is printed using the system printer device file. This value overrides the printer name specified in the print options associated with the document.

printer-device-file-name
Specify the name and library of the printer device file that is used for the print document request. This value overrides the printer file name specified in the print options associated with the document.

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 printer device file. If no library is specified as the current library for the job, QGPL is used.

library-name
Specify the name of the library where the printer device file is located.
File to receive output (OUTFILE)

Specifies the name of the database file in which the displayed information is stored. If the specified file does not exist, this command creates a database file and file member. If the file is created, the public authority for the file is the same as the authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library’s create authority. Output to OUTFILE is supported only if *OUTFILE is specified in the Display print options prompt (OPTIONS parameter).

*PRV The library and database file used in the previous (last) PRTDOC request for this user is used.

data-base-file-name

Specify the qualified name of the database file in which the displayed information is stored. If no file is found by that name, a file and member by that name are created and stored in the specified library, or in *CURLIB, if no library is specified.

The possible library values are:

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

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

library-name

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

Output member options (OUTMBR)

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

The possible member to receive output values are:

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

member-name

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

*PRV The member used in the previous (last) PRTDOC request for this user is used for this request.

The possible replace or add records values are:

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

*ADD The system adds the new records to the end of the existing records.
Current status value (CURSTS)

Specifies the value the document Interchange Document Profile (IDP) status field must have before the document may be printed to the database file. This field is 20 characters long and is valid only if OUTFILE output is requested.

*PRV The value used in the previous (last) PRTDOC request for this user is used.

*NOCHK The status field is not checked before printing this document to the database file.

value Specify the value that the status field must equal before the document is printed to the database file.

New status value (NEWSTS)

Specifies the value the document Interchange Document Profile (IDP) status field is set to after the document has been printed to the database file. If a value is specified on the New status value prompt (NEWSTS parameter), you must have at least *CHANGE authorization to the document. This field is 20 characters long and is valid only if OUTFILE output is requested.

*PRV The value used in the previous (last) PRTDOC request for this user is used.

*NOCHG The status field is not changed after printing this document to the database file.

value Specify the value to which the status field is set after the document is printed to the database file.

Type of data for output (OUTDTATYP)

Specifies whether the entire document, or only the Interchange Document Profile (IDP) information, is printed to the database file.

*PRV The value used in the previous (last) PRTDOC request for this user is used.

*ALL The entire document is printed to a database file.

*IDP Only the Interchange Document Profile (IDP) is printed to a database file.

Delete document (DLTDOC)

Specifies whether the document is deleted after it has been printed to the database file.

*NO The document is not deleted after being printed to the database file.

*YES The document is deleted after being printed to the database file.

Note: You must be the owner of the document or have *ALL authority to delete it.
Output device (OUTPUT)

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

*SAME
   The output device does not change.

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

* Your resolved document is shown on the display. A resolved document is a document with the text instructions processed.

Print device (DEV)

Specifies the name of the printer.

*SAME
   The printer does not change.

*USRPRF
   The printer ID specified in your user profile is used to print the document.

*SYSVAL
   The system printer is used to print the document.

*WRKSTN
   The printer assigned to the user’s work station is used to print the document.

printer-name
   Specify the name of the printer you want to use to print the document.

Output queue (OUTQ)

Specifies the name of the output queue. It must already exist.

*SAME
   The output queue does not change.

*DEV
   The output queue associated with the printer specified on the Print device prompt (DEV parameter) is used.

*FILE
   The output queue and output queue library values are taken from one of the following:
   1. If the Printer file prompt (PRTFILE parameter) is specified, values from the specified printer device file are used.
   2. If the Printer file prompt (PRTFILE parameter) is not specified, values from the Printer File Prompt on the document print options are used.

*WRKSTN
   The output queue assigned to the user’s work station is used.

output-queue-name
   Specify the name and library of the output queue that holds your output until it is ready to print.

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

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

Output file (SPLFILE)
Specifies the name of the output file.

*SAME
The output file does not change.

*FILE
The output file name is the name of the printer file used.

*DOC
The document name is used for the spooled file name. However, if the document name is longer than 10 characters, or contains a period, then the spooled file name is QSYSPRT.

spool-file-name
Specify the name of the file in which you want the output stored while it is on the output queue.

Form type (FORMTYPE)
Specifies the type of forms used in the printer.

*SAME
The forms type does not change.

*STD
The standard printer form for your computer system is used.

form-type
Specify the forms type for the type of paper on which the output is printed.

*BLANK
No special form type is specified.

Print separator page (COVERPAGE)
Specifies whether a cover page is printed that includes such things as the document name, folder name, document description, subject, reference, and author name.

*SAME
The cover page value does not change.

*YES
A cover page is printed.

*NO
A cover page is not printed.

Print quality (PRTQLTY)
Specifies the type of print quality that is used to print your document.
The print quality value does not change.

*LETTER
Your document is printed in letter quality type.

*TEXT
Your document is printed in text quality type. This is better quality than *DRAFT but not as good as *LETTER.

*DRAFT
Your document is printed in draft quality type.

Number of copies (COPIES)

Specifies the number of copies of your document you want to print.

*SAME
The copies value does not change.

value Specify a number, ranging from 1 through 99, for the number of copies of your document you want to print.

Print on both sides (DUPLEX)

Specifies whether output is printed on one side or two sides of the paper.

*SAME
The value does not change.

*YES The output is printed on both sides of the paper, with the top of each printed page at the same end of the sheet of paper.

*TUMBLE
The output is printed on both sides of the paper, with the top of one printed page at the opposite end from the top of the other printed page.

*NO The output is printed on one side of the paper.

Automatic page binding (AUTOBIND)

Specifies whether the left and right margins of alternating pages are adjusted to allow for page binding.

*SAME
The autobind option does not change.

*YES The margins are adjusted to allow for page binding.

*NO The margins are not adjusted to allow for page binding.
Delay printing (HOLD)

Specifies whether the printing of your documents is put on hold. The documents are held on the output queue, where you can release them to print, or delete them if you do not want them to print. You can print a group of documents together by putting them on the output queue before releasing them to print.

*SAME  The hold value does not change.
*YES   The printing is delayed for the specified documents.
*NO    Your documents begin printing when the printer is ready.

Print document error log (PRTERRLOG)

Specifies whether to include the document error log as part of the information printed with the document.

*PRV   The value used in the previous (last) PRTDOC request for this user is used for this request.
*YES   The error log is printed to the output device.
*NO    The error log is not printed to the output device.

Error log form type (ERRFORM)

Specifies the forms type for the type of paper on which the error log is printed.

*SAME  The error form value does not change.
*STD    The error log is printed on the paper specified in the printer file for the printer you selected.
error-form-name

Specify the name of the forms on which the error log is printed.

*BLANK No special error log form type is specified.

Large print (LARGEPRINT)

Specifies whether your document is printed using large print.

*SAME  The large print value does not change.
*YES   Your document is printed using large print.
*NO    Your document is not printed using large print.
**Merge type (MRGTYPE)**

Specifies where data being merged is stored.

*SAME
  The merge source does not change.

*QRY
  Data requested in a query is merged. A query is a request to select and copy one or more records from a file based on defined conditions.

*DOC
  Data stored in a document is merged.

*FILE
  Data stored in a file is merged.

*BLANK
  No data will be merged.

---

**Query (QRYDFN)**

Specifies the name of the query that is run to pass the data being merged. A query is a request to select and copy one or more records from a file based on defined conditions. *QRY must be specified on the Merge type prompt (MRGTYPE parameter).

*SAME
  The query name does not change.

query-definition-name
  Specify the name of the query that is used to move the data being merged.

The possible library values are:

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

library-name
  Specify the name of the library where the query is located.

---

**Data document (DTADOC)**

Specifies the name of the document that contains the data being merged. *DOC must be specified on the Merge type prompt (MRGTYPE parameter).

*SAME
  The document name does not change.

document-name
  Specify the name of the document using 1 to 12 alphanumeric characters. If you use more than 8 characters, the ninth character must be a period (.) followed by a 1- to 3-character extension.

---

**Data folder (DTAFLR)**

Specifies the name of the folder that contains the document being merged. *DOC must be specified on the Merge type prompt (MRGTYPE parameter).
*SAME
The folder name does not change.

folder-name
Specify the name of the folder that contains the document being merged.

---

Data file (DTAFILE)

Specifies the name of the file in which the member that contains data to be merged is located. *FILE must be specified on the Merge type prompt (MRGTYPE parameter).

*SAME
The data file name does not change.

file-name
Specify the name of the file that contains the data to be merged.

The possible library values are:

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

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

---

Data member (DTAMBR)

Specifies the name of the file member that contains the data to be merged. This parameter is valid only when MRGTYPE(*FILE) is specified.

*SAME
The file member does not change.

*FIRST
The first member in the file contains the data to be merged.

*FILE
The member with the same name as the file contains the data to be merged.

*LAST
The last member in the file contains the data to be merged.

member-name
Specify the name of the file member that contains the data being merged. A member is a set of data within a file.

---

Multiple line report (MLTLINRPT)

Specifies whether a multiple line report is created. A multiple line report is created by merging data field instructions. This creates a report in which each record of data produces several lines of output.

*SAME
The multiple line report option does not change.

*YES
A multiple line report is created.

*NO
A multiple line report is not created.
Adjust line endings (ADJLINES)

Specifies whether the line endings in the printed document are adjusted. The lines are adjusted according to what is specified on the Line Spacing/Justification display. This is useful when you print a document that has data merged into it, has instructions, has display attributes that do not print as spaces, or uses a proportionally spaced font.

*SAME
The line endings values do not change.

*YES  Adjusts line endings in the printed document.

*NO  Does not adjust the line endings in the printed document. This is useful if you have typed text exactly as you want it printed.

Adjust page endings (ADJPAGES)

Specifies whether the page endings in the printed document are adjusted. The pages are determined by what is specified for the first typing line and last typing line prompts on the Page Layout/Paper Options display.

*SAME
The page endings value does not change.

*YES  Page endings in the printed document are adjusted.

*NO  Page endings in the printed document are not adjusted.

Allow widow lines (ALWWIDOW)

Specifies whether the page endings are determined by the exact number of lines per page specified on the Page Layout/Paper Options display.

*SAME
The allow widow lines value specified in the print document options does not change.

*YES  Page endings are determined by the exact number of lines per page.

*NO  Page endings are not determined by the exact number of lines per page.

Renumber system page numbers (RENUMBER)

Specifies whether the page numbers are renumbered when the document is printed.

*SAME
The value does not change.

*YES  The page numbers are renumbered when the document is printed.

*NO  The page numbers are not renumbered when the document is printed.
Print change symbols (PRTCHGSYM)

Specifies whether change symbols are printed in the left margin on your document. Change symbols are used to indicate lines that have been revised.

*SAME  Print change symbol value does not change.
*YES   Change symbols are printed in the left margin of your document.
*NO    The change symbols are not printed in the left margin of your document.

Change symbols to print (SYMBOLS)

Specifies that up to 5 change symbol characters may appear in the left margin of the printed document. If your document contains more than one change symbol character and you do not select which change symbol characters you want to print, all change symbol characters specified in your document are printed.

*SAME  The change symbol value does not change.

value  Specify up to 5 change symbol characters to appear in the left margin of the printed document.

Draft spacing (DRAFTSPACE)

Specifies whether the spacing value can be adjusted for your document. For example, if the Line spacing prompt is 3 (Triple), then the doubled spacing value is 6, and five blank lines are printed between each line of text in your document. The document is still paginated using the value in the Line spacing prompt; so, depending on the amount of text on a page, one page may print over two pages.

*SAME  The draft spacing value does not change.
*YES   The spacing value for your document is doubled.
*NO    The spacing value that exists in the Line spacing prompt on the Line Spacing/Justification display is used.

Print line numbers (LINNBR)

Specifies whether line numbers are printed in your document. The line numbers begin with 1 on the first page of your document. Line numbers are not printed in headers or footers.

*SAME  The line numbers value does not change.
*YES   Line numbers are printed in your document.
*NO    Line numbers are not printed in your document.
Resolve instructions (RESOLVE)

Specifies whether the instructions that you have placed in your document are processed. For example, the Date instruction (.date) is resolved to the actual date (04/03/62).

*SAME  The resolve value does not change.
*YES   The instructions you have placed in your document are processed.
*NO    The instructions you have placed in your document are not processed. For example, the Date instruction (.date) is printed as *date.

Additional spaces to left (LEFTSPACES)

Specifies whether the left margin is increased.

*SAME  The left spaces value does not change.

value  Specify a number, ranging from 0 through 99, for the number of spaces that are added to the left margin in your printed document.

Character identifier (CHRID)

Specifies the graphic character set ID that is used to print your job. A graphic character set ID is an identifier that is used to specify a set of graphic characters in a code page. The graphic character set ID selected overrides the automatic value set by the system for the specific printer.

*SAME  The character set code page value does not change.

character-set  Specify the character set that is used to print your job. Up to 4 digits can be specified for the character set.

code-page  A code page is a particular assignment of hexadecimal identifiers to graphic characters. Up to 4 digits can be specified for the code page.

*BLANK  No special character set code page is used.

Save resolved output (SAVOUTPUT)

Specifies whether the document you are printing is also saved as a final form document.

*SAME  The save resolved output value does not change.
*YES   The printed document is saved as a final form document.
*NO  The printed document is not saved as a final form document.

Resolved output document (SAVDOC)
Specifies the name of the document that contains the final form document.

*SAME  The save document name does not change.

document-name  Specify the name of the document that contains the final form document. The document name ranges from 1 through 12 alphanumeric characters. If you use more than 8 characters, the ninth character must be a period (.) followed by a 1- to 3-character extension. If the document name you specify does not already exist, the document is created for you.

*BLANK  A resolved output document is not specified.

Resolved output folder (SAVFLR)
Specifies the name of the folder that contains the document being saved in final form.

*SAME  The save folder value does not change.

folder-name  Specify the name of the folder that contains the document.

*BLANK  A resolved output folder is not specified.

Place on job queue (JOBQ)
Specifies whether the print request is put on the job queue.

*SAME  The job queue value does not change.

*YES  The printing of the document is placed on the job queue.

*NO  The printing of the document is not placed on the job queue.

Job description (JOBD)
Specifies the name of the job description that describes how the job is run.

*SAME  The job description value does not change.

job-description-name  Specify the name of the job description that describes how the job is run.
The possible library values are:

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

library-name
Specify the name of the library where the job description is located.

Send completion message (SNDMSG)
Specifies whether you are putting your print job on the job queue and want a message sent to you when the job has completed.

*SAME  The send message value does not change.
*YES   A message is sent to you when the print job has completed.
*NO    A message is not sent to you when the print job has completed.

Cancel on error (CNLERR)
Specifies whether printing is stopped on your document if an error is detected.

*SAME  The cancel error value does not change.
*YES   Printing is stopped on your document if an error is detected. The error is listed in the error log with an error message stating that the job is canceled.
*NO    Printing continues on your document even if an error is detected.

Start page (STRPAGE)
Specifies the page number on which you want printing to start.

Note: If the STRPAGE(page-number) value specified is larger than the ENDPAGE(page-number) value specified, the entire document is printed.

*PAGERANGE  The pages specified on the PAGERANGE parameter are printed.
*SAME  The start page value does not change.
*FIRST  Printing is started on the first page of the document.
*LAST   Printing is started on the last page of the document.

page-number
Specify the page on which to begin printing. Valid values range from 0.01 through 9999.99.
End page (ENDPAGE)

Specifies the page number on which you want printing to stop.

*PAGERANGE

The pages specified on the PAGERANGE parameter are printed.

*SAME

The end page value does not change.

*FIRST

Printing is ended after the first page of the document.

*LAST

Printing is ended after the last page of the document.

*STRPAGE

The end page value is the same as the start page value. Only one page is printed.

\textit{page-number}

Specify the page on which to stop printing. Valid values range from 0.01 through 9999.99.

Page ranges (PAGERANGE)

Specifies the page ranges to print. A maximum of 7 ranges can be specified.

Note: If the STRPAGE\textit{(page-number)} value specified is larger than the ENDPAGE\textit{(page-number)} value specified, the entire document is printed.

*SAME

The page range specified on the document print options is printed.

The possible start page values are:

*FIRST

Printing is started on the first page of the document.

*LAST

Printing is started on the last page of the document.

\textit{page-number}

Specify the page on which to begin printing. Valid values range from 0.01 through 9999.99.

The possible end page values are:

*FIRST

Printing is ended after the first page of the document.

*LAST

Printing is ended after the last page of the document.

*STRPAGE

The end page value is the same as the start page value. Only one page is printed.

\textit{page-number}

Specify the page on which to stop printing. Valid values range from 0.01 through 9999.99.
Number of labels across page (LBLACROSS)

Specifies the number of labels that are printed across a page.

*SAME  The label across page value does not change.

value  Specify the number of labels you want printed across a page. Valid values range from 1 through 99.

Width of labels (LBLWIDTH)

Specifies the width (in number of characters) of the label. The width of a label is the number of characters from the left edge of the first label to the left edge of the next label, including the blank spaces between the labels. If the width you specify is larger than the margins for your document, the margins are used as the width.

*SAME  The label width value does not change.

value  Specify the width (in number of characters) that you want the label to be. Valid values range from 2 through 198.

Sheet feed labels (SHEETFEED)

Specifies, if you are sheet feed printing, whether you want more than one row of labels on a page. If you are using sheet feed paper, there is no other way to print more than one row of labels on a page.

*SAME  The sheet feed value does not change.

*YES  You are sheet feed printing and want more than one row of labels on a page.

*NO  You are not sheet feed printing, or you only want to print one row of labels on a page.

Number of rows per sheet (LBLDOWN)

Specifies, if *YES was selected for the Sheet feed labels prompt (SHEETFEED parameter), the number of rows of labels that you want printed on a page.

*SAME  The label down value does not change.

value  Specify the number of rows of labels that you want printed on a page. Valid values range from 1 through 99.

Shift left margin (SHFLEFTMAR)

Specifies whether to shift the left margin to prevent text from being truncated.
*SAME  The SHFLEFTMAR value does not change.

*YES  When the text exceeds the right margin or the paper edge, the left margin is shifted so that as much text as possible is printed. If the text does not exceed the right margin or the paper edge, the text is not shifted.

*NO  The left margin is not shifted when text exceeds the right margin. Any text exceeding the right margin is truncated.

---

**Examples**

**Example 1: Printing to a File**

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)  OPTIONS(*OUTFILE)
  OUTFILE(MYFILE/MYLIB)  OUTMBR(MYMBR *REPLACE)
  CURSTS(*PRV)  NEWSTS(*PRV)
  OUTDATATYP(*PRV)  PRTERRLOG(*PRV)  DLTOC(*NO)
```

This command prints the document MYDOC in folder MYFLR to the database file MYFILE in library MYLIB in the database file member MYMBR. If the member already exists, it is replaced by the contents of MYDOC. The CURSTS, NEWSTS, OUTDATATYP, and PRTERRLOG are taken from the last PRTDOC request. The document is not deleted after it is printed to the database file MYFILE.

**Example 2: Printing a Document**

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)  OPTIONS(*NO)
  DEV(MYPRNTR)  OUTQ(*DEV)
```

This command prints the document MYDOC in the folder MYFLR on a printer called MYPRNTR.

**Example 3: Printing Document Error Log**

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)  OPTIONS(*NO)  PRTERRLOG(*YES)
```

This command prints the document with a document error log attached to it.

**Example 4: Increasing Margin**

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)  OPTIONS(*NO)  LEFTSPACES(10)
```

This command prints the document and has 10 extra spaces inserted in the left margin.

**Example 5: Printing a Cover Page**

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)  OPTIONS(*NO)  COVERPAGE(*YES)
```

This command prints the document with a cover page.

**Example 6: Printing One Page to a File**

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)
  OPTIONS(*OUTFILE)  OUTFILE(MYLIB/MYFILE)
  OUTMBR(*FIRST)  PAGERANGE((5 5))
```

This command prints page 5 of the document to the database file MYFILE in library MYLIB in the first member.
Error messages

*ESCAPE Messages

CPF6C01  Error occurred during data stream transformation.

CPF6C03  Error occurred during document conversion.

CPF9012  Start of document interchange session not successful for &1.

CPF9801  Object &2 in library &3 not found.

CPF9810  Library &1 not found.

CPF9820  Not authorized to use library &1.

OFCFFFFC User storage capacity exceeded.

OFCFFFFD Damaged object found.

OFC8EA3  OfficeVision for AS/400 editor is not available to resolve to a display.

OFC8E01  Printer ID selected is not correct.

OFC8E1C  Cannot delay output when spooling is not active.

OFC8E1D  Printer for large print is not correct.

OFC8E2A  Output file member is in use.

OFC8E2B  Not authorized to output disk file or library.

OFC8E2C  Output disk file member could not be opened.

OFC8E30  Incorrect character set ID specified.

OFC8E38  Member is not a document output file member.

OFC8E4D  Output file name &9 is incorrect.

OFC8E50  Job has been canceled because of error.

OFC8E6B  Not authorized to output disk file member.

OFC8E6D  Could not access the output disk file member.
OFC80B5
OfficeVision for OS/400 editor is not available on the system.

OFC800A
Folder is in use.

OFC800B
Document &1 is in use.

OFC800E
&1 already exists as document or folder.

OFC800F
Display does not support text.

OFC8006
Folder not found.

OFC8007
Document &1 not found in folder.

OFC8008
Request not allowed with folder.

OFC8009
Request not allowed with document &1.

OFC801A
Document has been saved to diskette, tape or save file.

OFC8010
Document &1 cannot be processed.

OFC8011
Document &1 needs to be recovered.

OFC8016
Document &1 is checked out.

OFC8018
Document &1 is empty.

OFC802C
Label option specified with non-label document.

OFC802D
Option not allowed for PC editor.

OFC8029
Cannot save resolved output when printing a resolved document.

OFC820D
Library &4 was not found.

OFC820F
Member &3 is in use.

OFC947E
Fill-in document &1 could not be opened.

OFC9486
Printer file or printer file library was not found.

OFC960A
&1 key was pressed by the user to end the PRTDOC function.
OFC9609
   &1 is the resolved output file name for the print options function.

OFC980B
   &9 documents printed, &10 documents not processed.

OFC980C
   Error printing document &1 to a file.

OFC980D
   Error converting document &1.

OFC980E
   Error converting document &1.

OFC980F
   Could not delete document &1 from folder.

OFC9801
   Document &1 could not be opened.

OFC9802
   Folder could not be opened.

OFC9806
   No documents were selected for printing.

OFC9808
   Document &1 does not have selected status.

OFC9809
   Error log incorrect with document descriptions only.

OFC9810
   Could not update status for document &1.

OFC9811
   Folder needs to be reclaimed.
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPTH (Merge TCP/IP Host Table)
The Print Disk Information (PRTDSKINF) command is used to print disk space information that was stored in database file QAEZDISK or QAEZDnnnnn by the Retrieve Disk Information (RTVDSKINF) command, where 'nnnnn' is the ASP number of the independent ASP for which disk space information was retrieved. The output with file name QPEZDISK goes to the spool queue associated with the job using this command.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPTTYPE</td>
<td>Type of report</td>
<td>*LIB, *FLR, *OWN, *OBJ, *SYS</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>ASPDEV</td>
<td>ASP device</td>
<td>Name, *SYSBAS</td>
<td>Optional</td>
</tr>
<tr>
<td>LIB</td>
<td>Libraries</td>
<td>Generic name, name, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>OWNER</td>
<td>Owners</td>
<td>Generic name, name, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>FLR</td>
<td>Folders</td>
<td>Generic name, name, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>DOC</td>
<td>Documents</td>
<td>Generic name, name, *ALL, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>OBJ</td>
<td>Objects</td>
<td>Generic name, name, *ALL, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>MINSIZE</td>
<td>Smallest size</td>
<td>Decimal number, 0</td>
<td>Optional</td>
</tr>
</tbody>
</table>
**Type of report (RPTTYPE)**

Specifies the type of report to print. The report information is taken from member QCURRENT in QAEZDISK or QAEZDnnnnn, where ‘nnnnn’ is the ASP number of the independent ASP for which disk space information was retrieved. If QCURRENT does not contain any data, an error message is sent.

This is a required parameter.

**Restrictions:**

- If option *FLR is specified for the RPTTYPE parameter, *SYSBAS must be specified for the ASPDEV parameter. Folders are not allowed on auxiliary storage pool (ASP) devices, they are only allowed on the system ASP and basic ASPs.

  *LIB  A report of the library information contained in the file is printed.

  *FLR  A report of the folder information contained in the file is printed.

  *OWN  A report of the user profile (owner) information contained in the file is printed.

  *OBJ  A report of object information contained in the file is printed.

  *SYS  A report of only the system information contained in the file is printed.

**ASP device (ASPDEV)**

Specifies the auxiliary storage pool (ASP) device for which disk space information is to be printed.

  *SYSBAS  Disk information for the system ASP and all basic ASPs is printed. File QAEZDISK in library QUSRSYS contains the disk space information that is to be printed.

  name  Specify the name of the ASP device for which disk space information is to be printed. File QAEZDnnnnn in library QUSRSYS contains the disk space information that is to be printed, where ‘nnnnn’ is the ASP number of the specified ASP device.

**Libraries (LIB)**

Specifies the names of the libraries to print information about.

  *ALL  The report has information on all user libraries on the system.

  name  Specify the user library.

  generic-name  Specify the generic library name.

**Owners (OWNER)**

Specifies the names of the owners (user profiles) to print information about.

  *ALL  The report contains information on all user profiles on the system.

  name  Specify the name of a user profile.
**generic-name**

Specify the generic user profile name.

---

**Folders (FLR)**

Specifies the names of the folders to print information about.

- **ALL** The report has information on all user folders on the system.
- **name** Specify the folder name.
- **generic-name** Specify the generic folder name.

---

**Documents (DOC)**

Specifies the names of the documents to print information about.

- **ALL** The report contains information on all documents in the specified folder.
- **name** Specify the document by the given name within the specified folder.
- **generic-name** Specify the documents specified by the generic qualification.

---

**Objects (OBJ)**

Specifies the names of the objects to print information about.

- **ALL** If you specify a library or owner, then the object information is all objects within the library or those controlled by the owner.
- **NONE** No library or owner is specified.
- **name** Specify a library or owner, then the object information is the object specified by the given name within the library or controlled by the owner.
- **generic-name** Specify a library or owner, then the object information are the objects that meet the specified generic qualification within the library or controlled by the owner.

---

**Object types (OBJTYPE)**

Specifies the object types to print information about.

**Single values**

- **ALL** If you specify a library or owner, information is printed on all the specified object types within the library or controlled by the owner. If an object name is specified, information on all object types with that name, within the library, or controlled by the owner is printed. If a library or
owner is not specified, the report has information on all object types on the system. If an object name is specified, information only on object types with that name is printed.

Other values (up to 60 repetitions)

**object-type**
Specify a library or owner, then the object type information is the object type specified within the library or controlled by the owner. If an object is specified, the report has information on the objects with the specified object type within the library or controlled by the owner.

---

**Smallest size (MINSIZE)**

Specifies the size of the smallest piece of information to include. For example, if a library report is requested without objects, then this size would be the size of the smallest library to include. If objects within the library are requested, then this would be the size of the smallest object within the library to include.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>All objects are included regardless of size.</td>
</tr>
<tr>
<td><em>size</em></td>
<td>Specify size in thousands of bytes.</td>
</tr>
</tbody>
</table>

---

**Sort by (SORT)**

Specifies the order in which the information should be sorted.

**SIZE**  Information is sorted from large to small.

**OWNER**  The information is sorted in alphabetical order by owner name.

**LSTCHG**  The information is sorted by last-change date with the oldest information first.

**LSTUSE**  The information is sorted by last-use date with the oldest information first.

**NAME**  Information is sorted in alphabetical order according to the report type.

---

**Examples**

PRTDSKINF ASPDEV(*SYSBAS) RPTTYPE(*LIB) LIB(*ALL) OBJ(*ALL) SORT(*SIZE)

This command prints a library report from database file QAEZDISK in library QUSRSYS in member QCURRENT, containing information about all libraries, objects, and object types in the libraries. The information is sorted by size and sent to the printer file QPEZDISK.
Error messages

*ESCAPE Messages

CPF1ED0
    Current collection of disk space information not found.

CPF1ED1
    Not authorized to collect disk space information.

CPF1ED2
    File &1 is in use and cannot be accessed.

CPF1EEC
    Not authorized to file &1.

CPF1E99
    Unexpected error occurred.
Print Error Log (PRTERRLOG)

Where allowed to run: All environments (*ALL)

Threadsafe: No

The Print Error Log (PRTERRLOG) command is used primarily for problem analysis tasks. It places a formatted printer file of the data in the machine error log in a spooled printer device file named QPCSMPRT or in a specified output file.

Restrictions:
- The following user profiles have private authorities to use the command:
  - QPGMR
  - QSYSOPR
  - QSRV
  - QSRVBAS

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV</td>
<td>Logical device</td>
<td>Single values: *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values (up to 10 repetitions): Name</td>
<td></td>
</tr>
<tr>
<td>RSRCNAME</td>
<td>Resource name</td>
<td>Values (up to 10 repetitions): Name</td>
<td>Optional</td>
</tr>
<tr>
<td>ERRLOGID</td>
<td>Error log identifier</td>
<td>Values (up to 10 repetitions): Hexadecimal value</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Output</td>
<td>*PRINT, *OUTFILE</td>
<td>Optional</td>
</tr>
<tr>
<td>PERIOD</td>
<td>Time period for log output</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1:</td>
<td>Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Beginning time</td>
<td>Time, *AVAIL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Beginning date</td>
<td>Date, *CURRENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2:</td>
<td>Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Ending time</td>
<td>Time, *AVAIL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Ending date</td>
<td>Date, *CURRENT</td>
<td></td>
</tr>
<tr>
<td>PRTFMT</td>
<td>Print format</td>
<td>*CHAR, *HEX</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTFILE</td>
<td>File to receive output</td>
<td>Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: File to receive output</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>OUTMBR</td>
<td>Output member options</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Member to receive output</td>
<td>Name, *FIRST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Replace or add records</td>
<td>*REPLACE, *ADD</td>
<td></td>
</tr>
<tr>
<td>VOLTYPE</td>
<td>Volume type</td>
<td>Character value</td>
<td>Optional</td>
</tr>
<tr>
<td>MODEL</td>
<td>Model number</td>
<td>Character value</td>
<td>Optional</td>
</tr>
</tbody>
</table>
### Type of log data to list (TYPE)

Specifies the type of error log data from the machine error log to print in the spooled printer file.

- **ALL**  
  All the error codes in the machine’s error log are printed. In addition, the error codes for each subsystem (for example, diskette units, printers) are printed in summary form.

- **ALLSUM**  
  All the data in the error log is printed in summary form.

- **ANZLOG**  
  A one-line summary is created for each entry in the error log.

- **MCH**  
  Only the error data produced by machine checks is printed.

- **DEV**  
  Only the error data produced by the devices specified in the following parameters is printed:
    - Logical device (DEV) parameter
    - Resource name (RSRCNAME) parameter

- **ERRLOGID**  
  Only the error data with the specified error log record is printed. *ERRLOGID* can only be specified if the Error log identifier (ERRLOGID) parameter is also specified. It is ignored for other request types.

- **VOLSTAT**  
  Only the tape or diskette volume statistical data records are printed.

**Note:** If you specify *PRINT on the Output (OUTPUT) parameter and *VOLSTAT on this parameter, lifetime statistics are printed. If you specify *OUTFILE on the OUTPUT parameter and *VOLSTAT on this parameter, session statistics are directed to the output file. If the name of the volume is reported as ‘******’, it means that this volume is not displayable.

### Logical device (DEV)

Specifies the device names for which you want the error log data to be printed. This parameter is valid only if *DEV is specified for the Type of log data to list (TYPE) parameter. This parameter cannot be specified if a value is specified for the Resource name (RSRCNAME) parameter.

**Single values**

- **ALL**  
  The error log data for all device names is printed.

**Other values**

---

Top
name Specify one or more device names whose error log data you want to print. A maximum of ten device names can be specified.

**Resource name (RSRCNAME)**

Specifies the resource names for which error log entries are to be printed. This parameter is valid only if *DEV is specified for the Type of log data to list (TYPE) parameter. This parameter cannot be specified if a value is specified for the Logical device (DEV) parameter.

name Specify one or more resource names whose error log data you want to print. A maximum of ten resource names can be specified.

Note: If you specify a storage controller input/output processor (IOP) as the resource name, no error log entries are printed for the resource.

**Error log identifier (ERRLOGID)**

Specifies that error log entries with the specified error log identifier are printed. This parameter is valid only if *ERRLOGID is specified for the Type of log data to list (TYPE) parameter. A maximum of ten error log identifiers can be specified.

hexadecimal-value

Specify the error log identifier of an error log entry to be printed.

**Output (OUTPUT)**

Specifies whether the output from the command is printed with the job’s spooled output or sent to a database file.

*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.

**Time period for log output (PERIOD)**

Specifies the period of time for which the error log data is printed. The following values can be coded in this parameter, which contains two sets of two values each.

Note: This parameter is not valid when TYPE(*VOLSTAT) and VOLSTATYP(*LIFETIME) are specified.

Element 1:

Element 1: Beginning time
The error data that is available for the specified start or end date is printed.

**time** Specify the start time of the specified start date for which the error data is printed. 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.

**Element 2: Beginning date**

**CURRENT**

The error data that is available for the current day and between the specified start and end times (if specified) is printed.

**date** Specify the start date for which error data is printed. The date must be specified in the job date format.

**Element 2:**

**Element 1: Ending time**

**AVAIL**

The error data that is available for the specified start or end date is printed.

**time** Specify the end time for the specified end date that specifies the error data to be printed. See the **Beginning time** description on this parameter for details about how time must be specified.

**Element 2: Ending date**

**CURRENT**

The error data that is available for the current day and between the specified start and end times (if specified) is printed.

**date** Specify the end date for which error data is printed. The date must be specified in the job date format.

---

**Print format (PRTFMT parameter)**

Specifies whether the indicated report prints any hexadecimal data in character format. This parameter cannot be specified if *VOLSTAT* is specified for the **Type of log data to list (TYPE)** parameter, or if a value is specified for the **File to receive output (OUTFILE)** parameter.
The report is formatted so that hexadecimal data prints as character data.

No formatting is done for the report. Hexadecimal data prints as hexadecimal.

---

**File to receive output (OUTFILE)**

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

**Qualifier 1: File to receive output**

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

**Qualifier 2: Library**

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

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

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

---

**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.
Volume type (VOLTYPE)
Specifies the volume type of the specified volume identifier. Valid entries are 4-digit device type numbers for cartridge tape, reel tape, or diskette. This parameter returns information about all the volumes that use the same technology as the tape device type that was specified. For example, if 6380 is the specified value for this parameter, information about all 1/4 inch tape cartridges on the system is returned.

*character-value
Specify the volume type.

Model number (MODEL)
Specifies the model number of the specified model type. This parameter is required if the device type is 9331 and TYPE(*VOLSTAT) is specified.

*character-value
Specify the model number.

Volume (VOL)
Specifies the name of the volume for which you want statistics processed.

Single values

*ALL Volume statistics are processed for all volumes.

Other values

*character-value
Specify the name of the volume for which statistics are processed. A maximum of ten volume names can be specified.

Volume statistical data (VOLSTAT)
Specifies whether the volume statistical data records are kept or deleted from the machine error log after they are printed. This parameter is valid only if *VOLSTAT is specified on the Type of data (TYPE) parameter.

Note: ENDOPT(*UNLOAD) must be specified during the SAVE operation to generate volume statistics at the completion of the tape operation.

*KEEP The volume statistical data records are kept in the error log after they are printed.

*DLT The volume statistical data records are deleted from the error log for volumes that are not active after they are printed.

Notes:
1. You cannot specify *DLT on this parameter if *OUTFILE is specified on the Output (OUTPUT) parameter.
2. The length of time it takes to run this command when VOLSTAT(*DLT) is specified is dependent on the number of volume IDs being deleted.

**Volume statistics type (VOLSTATTTYP)**

Specifies the type of volume statistics printed or directed to an output file. This parameter is valid only if *VOLSTAT* is specified on the **Type of log data to list (TYPE)** parameter.

* **LIFETIME**
  
  Lifetime statistics are printed. Lifetime statistics cannot be placed in an output file.

* **SESSION**
  
  Session statistics are directed to the output file specified on the **File to receive output (OUTFILE)** parameter. Session statistics cannot be printed.

**Error log entries to select (SELECT)**

Specifies which type of error log entries are included on the report.

* **ALL**
  
  All error log entries are included on the report.

* **PRC**
  
  The processor error log entries are included on the report.

* **MEDIA**
  
  The error log entries for disk, tape, and diskette devices are included on the report.

* **LWS**
  
  The error log entries for local workstations are included on the report.

* **CMN**
  
  The error log entries for communications are included on the report. These include entries for communications I/O processors, I/O adapters, ports, lines, controllers, and devices connected with SDLC, ASYNC, BSC, X.25, IDLC, ISDN, and local area network line protocols.

* **PWR**
  
  The error log entries for system power control network (SPCN) are included on the report.

* **LPP**
  
  The error log entries for licensed programs are included on the report.

* **LIC**
  
  The error log entries for Licensed Internal Code are included on the report.

**Sort by (SORT)**

Specifies the order in which the entries appear on the report.

* **DATETIME**
  
  The entries are sorted by date and time. The summary entries are for each day.

* **TIME**
  
  The entries are sorted by the time of day only. The summary entries are for each hour.

* **DEVADR**
  
  The entries are sorted by the address of the device. The summary entries are divided into three levels: those for which the first two digits of the address are the same, those for which the first four digits of the address are the same, and those for which the first eight digits of the address are the same.
*ERRTYPE
The entries are sorted by the severity of the type of error. The more severe types of errors report at the top of the list. The summary entries are divided into two levels: those that have a common error type, and those that have a common error type and system reference code.

*RSRCNAME
The entries are sorted by the resource name of the device.

Examples

Example 1: Printing Error Log Data
PRTERRLOG

This command gets the error data in the machine error log that occurred for all device types and puts it in a spooled file. The entire error log is printed and any hexadecimal data is in character format.

Example 2: Using the System Resource Manager Database
PRTERRLOG TYPE(*DEV) RSRCNAME(TAPE000001) PRTFMT(*HEX)

This command uses the system resource manager database to determine the device type, model, and serial number for the resource TAPE000001. The print request is based on that information. The report is put in the spooled file and contains all records that pertain to that device type, model, and serial number. Any hexadecimal data in the file is converted to hexadecimal format.

Example 3: Processing Error Log Entries
PRTERRLOG TYPE(*DEV) DEV(DISKLUD1) OUTPUT(*OUTFILE)
OUTFILE(MYLIB/MYDBD) OUTMBR(ELOG)

This command processes all the error log entries for the logical device named DISKLUD1. They are put in the file MYDBD, in the library MYLIB, and in the member ELOG. No spooled files are created.

Error messages

*ESCAPE Messages

CPD36CA
OUTPUT(*OUTFILE) cannot be specified with DEV(*ALL).

CPD36C2
DEV and RSRCNAME cannot be used together.

CPD36C3
PRTFMT parameter not valid with TYPE(*VOLSTAT).

CPD36C4
OUTFILE not valid with PRTFMT parameter.

CPD36C5
RSRCNAME parameter can only be used with TYPE(*DEV) parameter.

CPD36C7
ERRLOGID valid only with TYPE(*ERRLOGID).

CPD36C9
PERIOD not valid for specified TYPE and VOLSTATYP.
CPF3535
   Error log not available for printing.

CPF3541
   No error log entries were found.

CPF3593
   PERIOD parameter start time exceeds end time.

CPF3693
   Service function ended because error occurred.

CPI36CA
   Resource name &1 not found.

CPI36CC
   No error log entries were found for &1 &2.
The Print Internal Data (PRTINTDTA) command is used primarily for problem analysis tasks. It writes the machine internal data to a spooled printer file. The data is used to service the system.

Restrictions:
1. This command is shipped with public *EXCLUDE authority and the PGMGR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.
2. To use this command, you must have service (*SERVICE) special authority, or be authorized to the Service dump function of Operating System through iSeries Navigator’s Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_DUMP, can also be used to change the list of users that are allowed to perform dump operations.
3. The command must be issued from within the job with internal data being printed, or the issuer of the command must be running under a user profile which is the same as the job user identity of the job with internal data being printed, or the issuer of the command must be running under a user profile which has a job control (*JOBCTL) special authority. The job user identity is the name of the user profile by which a job is known to other jobs. It is described in more detail in the Work Management Guide.
4. This command is intended to be used only at the direction of your iSeries service representative.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>Type of data</td>
<td>*DMP, *INTCFG, *NOTES, *JOB</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>DMPID</td>
<td>Dump identifier</td>
<td>Character value, *NONE, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>PERIOD</td>
<td>Time period for internal data</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td>JOB</td>
<td>Job name</td>
<td>Single values: *, *SVRTYPE Other values: Qualified job name</td>
<td>Optional</td>
</tr>
<tr>
<td>SLTTHD</td>
<td>Thread ID to include</td>
<td>Single values: *ALL, *SELECT Other values (up to 20 repetitions): Hexadecimal value</td>
<td>Optional</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SVRTYPE</td>
<td>Server type</td>
<td>Single values: *NONE Other values (up to 5 repetitions): Generic name, name</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**Type of data (TYPE)**

Specifies the type of data to be printed.

This is a required parameter.

*DMP*  The data to print was dumped by a previously issued Dump Job Internal (DMPJOBINT) command or by the machine when it was processing a device error or object damage. The dump identifier of the data or *ALL must be specified for the Dump identifier (DMPID) parameter.

*INTCFG*  The machine internal configuration and resource information is printed.

*NOTES*  The notes portion of the machine internal data, for the period specified for the Time period for internal data (PERIOD) parameter, is printed.

*JOB*  The data to be printed is for the job specified for the JOB parameter.

**Dump identifier (DMPID)**

Specifies, for internal dumps only, the dump identifiers associated with the machine internal data that is printed. This parameter must be specified only if *DMP is specified on the Type of data (TYPE) parameter; otherwise, it is ignored.

*NONE*  No dump identifier is specified.

*ALL*  The dump portion of the machine internal data, for the period specified on the Time period for internal data (PERIOD) parameter, is printed.

*character-value*  Specify the dump identifier of the dump output that is printed. The identifier specified must contain 8 characters.

**Time period for internal data (PERIOD)**

Specifies the period of time for which the notes or dump portion of the machine internal data is printed. This parameter is valid only if *NOTES is specified for the Type of data (TYPE) parameter or if *DMP is specified on the TYPE parameter and *ALL is specified for the Dump identifier (DMPID) parameter; otherwise, it is ignored.

**Element 1: Start time and date**

**Element 1: Beginning time**
The notes or dump data that are available from the beginning date to the ending date (or for the current day only) are printed.

time Specify the beginning time for the specified beginning date for which you want the notes or dump data printed. 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.

date Specify the beginning date for which you want the notes or dump data printed. The job date format must be used.

date Specify the ending date for which you want the notes or dump data printed. The system date format must be used.
**Job name (JOB)**

Specifies the qualified name of the job for which the data will be dumped. This parameter must be specified **only** if *JOB* is specified for the **Type of data (TYPE)** parameter; otherwise, it is ignored.

**Single values**

* The job that issued this command is the job that will be dumped.
*SVRTYPE
  All jobs whose server type matches the server type attribute specified for the **Server type (SVRTYPE)** parameter will be dumped.

**Qualifier 1: Job name**

**name** Specify the name of the job to be dumped.

**Qualifier 2: User**

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

**Qualifier 3: Number**

**000000-999999**

Specify the system-assigned job number of the job to be dumped.

---

**Thread ID to include (SLTTHD)**

Specifies a list of up to twenty threads whose information is to be included. This parameter must be specified **only** if *JOB* is specified for the **Type of data (TYPE)** parameter; otherwise, it is ignored.

**Single values**

*ALL* All threads are dumped.

*SELECT
  A list of thread identifiers is shown from which the user can select up to twenty to be included.
  *SELECT is only valid if the PRTINTDTA command is run in an interactive session; otherwise, an error message is sent.

**Other values**

**thread-identifier**

Specify the identifiers of up to twenty threads whose information is to be included. A thread identifier is a string of eight hexadecimal characters.

---

**Server type (SVRTYPE)**

Specifies the server type attribute to identify the job to be dumped. This parameter must be specified **only** if *SVRTYPE* is specified for the **Job name (JOB)** parameter; otherwise, it is ignored. All jobs whose server types matches this value will be dumped. For a list of possible server types, see **Work Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter**

**Single values**

---

274 IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPTH (Merge TCP/IP Host Table)
*NONE
No server types are dumped.

Other values

generic-name
Specifies the generic server type used to identify the job to be dumped. 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. All jobs whose server type matches the specified generic prefix will be dumped.

name Specify the server type used to identify the job to be dumped. A list of up to five server types can be specified.

Examples

Example 1: Dump by Dump Identifier
PRTINTD TA TYPE(*DMP) DMPID(0102FA3C)

This command prints the job internal dump output that has a dump identifier of 0102FA3C.

Example 2: Dump by Job Identifier
PRTINTD TA TYPE(*JOB) JOB(201230/ALMATM/QPADEV0008) SLTTHD(*ALL)

This command prints the job internal dump output for the selected job including all threads information.

Example 3: Dump by Task Name
PRTINTD TA TYPE(*TASK) TASK(MSCP)

This command prints the dump output for the task named MSCP.

Example 4: Dump a Job by Specifying Server Type
PRTINTD TA TYPE(*JOB) JOB(*SVRTYPE) SVRTYPE(QIBM_FTP)

This command prints the dump output for the job with the server type set to QIBM_FTP.

Error messages

*ESCAPE Messages

CPF3517
Cannot specify *SELECT for the thread ID to include.

CPF3519
Cannot start service function.

CPF6801
Command prompting ended when user pressed &1.

CPF98A2
Not authorized to &1
Print IP over SNA (PRTIPSCFG)

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

The Print IP over SNA Configuration (PRTIPSCFG) command prints information about the current AF_INET Sockets over SNA configuration. The spooled file created by this CL command is named QSYSPRT. It is sent to the job default output queue. The user data value of the spooled file is PRTIPSCFG.

There are no parameters for this command.

Parameters
None

Examples
PRTIPSCFG

This command prints the current AF_INET sockets over SNA configuration data.

Error messages
*ESCAPE Messages
CPFA116
   &1 configuration not printed.

© Copyright IBM Corp. 1998, 2006
**Print JOBD Authority (PRTJOBDAUT)**

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

*Threadsafe:* No

The Print Job Description Authority (PRTJOBDAUT) command allows you to print a report of the job descriptions in a library that do not have public authority of *EXCLUDE, and a user name is specified in the job description. This is a way to check for job descriptions that every user on the system is authorized to use that allow the user to run as another user profile.

This command will print two reports for a library. The first report (Full Report) will contain all of the job descriptions that do not have public authority of *EXCLUDE and have a user name specified. The second report (Changed Report) will contain the job descriptions that *now* do not have public authority of *EXCLUDE or have a user name specified that either did have public authority of *EXCLUDE, did not have a user name specified, or did not exist when the PRTJOBDAUT command was previously run for the library. If the PRTJOBDAUT command was not previously run for the library, there will be no ‘Changed Report’. If the command has been previously run for the library but no additional job descriptions do not have public authority of *EXCLUDE and a user name specified, then the ‘Changed Report’ will be printed but there will be no job descriptions listed. Changes to user profile special authorities will not cause a ‘Changed Report’ to be generated.

The reports will contain the following information:

- The name of the library that was specified on the command.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each job description that does not have *PUBLIC authority of *EXCLUDE and has a user name specified. Each entry contains the following information:
  - The name of the library the job description is in.
  - The name of the job description.
  - The owner of the job description.
  - The name of the user profile specified in the job description.
  - The special authorities associated with the user profile. The special authorities that are shown are all of the special authorities that would be available when that job description is used. The special authorities shown are the special authorities that the user has, plus the special authorities that the user’s group profiles have (if the user has any groups).

The file QSECJBDOLD in library QUSRSLBL contains information from the last time the PRTJOBDAUT command was run for a library. There is a member within the file, with the same name as the library, for each library that has been previously specified on the command. If a special value is specified for the library name (for example, *USRLIBL), then the ‘*’ will be replaced with a ‘Q’ in the member name. System file QAOBJAUT in library QSYS with format name of QSYDSTAUT is the model file for the QSECJBDOLD file.

**Restriction:** You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.
Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHGPTONLY</td>
<td>Changed report only</td>
<td>*NO, *YES</td>
<td>Optional, Positional 2</td>
</tr>
</tbody>
</table>

Library (LIB)

This is a required parameter.

The name of the library to search for job descriptions with public authority that is not *EXCLUDE and a user name is specified.

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

*CURLIB  The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched. If the ASP device (ASPDEV) parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*USRLIBL  If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the ASP device (ASPDEV) parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*ALL  All the libraries in the auxiliary storage pools (ASPs) specified for the ASP device (ASPDEV) parameter are searched.

*ALLUSR  All user libraries in the auxiliary storage pools (ASPs) defined by the ASP device (ASPDEV) parameter are searched.

User libraries are all libraries with names that do not begin with the letter Q except for the following:

- #CGULIB
- #DSULIB
- #SEULIB
- #COBLIB
- #RPGLIB
- #DFULIB
- #DALIB

Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

- QDSNX
- QRLxxxxx
- QUSRJJS
- QUSRvxRxMx
- QGPL
- QSRVAGT
- QSRINFSKR
- QGPL3B
- QSY52
- QUSRNOTES
- QMGTC
- QSY52xxxxx
- QUSROND
- QMGTC2
- Q366F
- QUSRPOS6S
- QMPDATA
- QUSER3B
- QUSRPOSSA
- QMQMATA
- QUSRADSM
- QUSRPOSRA
- QMQMPROC
- QUSRBRM
- QUSRRDARS
- QPFRDATA
- QUSRDR1CL
- QUSRYS
- QRC1
- QUSRDR1DB
- QUSRVI

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

**ALLAVL**
All libraries in all available ASPs are searched.

**ALLUSRAVL**
All user libraries in all available ASPs are searched. Refer to *ALLUSR for a definition of user libraries.

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

---

**Changed report only (CHGRPTONLY)**

Specifies whether just the changed report should be printed.

*NO* The full and changed reports will be printed.

*YES* Only the changed report will be printed.

---

**Examples**

PRTJOBDAUT LIB(QGPL)

This command prints both full and changed report for the job descriptions in the library QGPL.

---

**Error messages**

**ESCAPE Messages**

CPFBB304
User does not have required special authorities.

CPFBB307
Command &1 in use in another job.
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPHT (Merge TCP/IP Host Table)
Print Profile Internals (PRTPRFINT)

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

The Print Profile Internals (PRTPRFINT) command allows you to print a report containing information on the number of entries contained in a user profile (*USRPRF) object. The number of entries in the user profile determines the size of the user profile.

Four types of entries can be found in a user profile:

**Owned objects**
For every object a user profile owns, an "owned objects" entry exists in that user's profile (*USRPRF).

**Private authorities**
For every private authority a user is granted, a "private authority" entry exists in that user's profile (*USRPRF).

**Authorized objects**
For every user that is granted a private authority to an object a profile owns, an "authorized object" entry exists in the object owner's profile (*USRPRF).

**Primary group authorities**
For every object for which a user is the primary group, a "primary group" entry exists in that user's profile (*USRPRF).

Each entry in the user profile causes the user profile (*USRPRF) object to grow. The combination of all of the entries determines the size of the user profile. A user profile (*USRPRF) can contain approximately 10 million entries for objects in the system auxiliary storage pool plus an additional 10 million entries for each independent auxiliary storage pool which is varied on that the profile owns objects on, has private authority to objects on, or is the primary group profile of objects on it. A user profile can not exceed 10 million entries for the objects on any single auxiliary storage pool.

You can also think of the total number of entries as determining how "full" a user profile is. The report produced by this command shows how full the user profile is by giving a percentage (rather than giving the actual number of entries in the profile.) The report also gives a percentage for each of the four types of entries in the *USRPRF. Note: In the report, the total percentage full for a profile can be greater than 100% due to rounding.

You can choose to run this report for all user profiles, a subset of profiles, a specific profile or all profiles that are at least a specified percentage full. For example, you can run the report for the CJW profile or you can run the report for all profiles that are at least 99.90% full.

**Note:** If your system has any independent auxiliary storage pools varied on, then the percentages produced by this report may not be what you expect. The percentage is computed for each profile based on the total number of entries used divided by the total number of entries available to that specific profile. The total number of entries available to each profile may not be the same depending on whether the profile has entries for any object on a varied on independent auxiliary storage pool. For example, if the system has two varied on independent auxiliary storage pools and profile TESTUSER1 only has entries for objects on one of them, then the total number of entries available for TESTUSER1 is 20 million. If profile TESTUSER2 has entries for objects on both of the independent auxiliary pools, then the total...
number of entries available for TESTUSER2 is 30 million. However, if both of the independent auxiliary storage pools are varied off, then the total number of entries available for profiles TESTUSER1 and TESTUSER2 is 10 million.

Recommendations to avoid profiles becoming full:

- Do not have one profile own everything on your system. For example, have each application be owned by its own profile.
- Do not use IBM-supplied profiles, such as QSECOFR and QPGMR, as owners of your application. As shipped from IBM, they already own many objects and can become full when they also own user (non-IBM) objects.
- If you are granting private authorities to many objects for several users, you should consider using an authorization list to secure the objects. Authorization lists will cause one private authority entry for the authorization list in the user’s profile rather than one private authority entry for each object. In the object owner’s profile, authorization lists will cause an authorized object entry for every user granted authority to the authorization list rather than an authorized object entry for every object multiplied by the number of users granted the private authority.

Authorization lists are especially useful if you are granting private authorities to files. Files are complex objects. For complex objects, you get an entry for each piece of the object. For example, in a file owner’s profile, you have an ownership entry for each piece of the file, including an entry or two for each member. (Physical files have two entries per member.) If you grant a private authority to ten users and the file has 50 members, the result will be 100 authorized object entries in the file owner’s profile. With an authorization list, the ownership entries will remain the same, but the authorized object entries will be reduced to one for each user granted authority to the authorization list securing the file.

Do not confuse the percentage full of a user profile with the maximum storage (MAXSTG) that a user profile can own. They are two different concepts.

**Restriction:** You must have all object (*ALLOBJ) special authority to run this command.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>Select by</td>
<td>*USRPRF, *PCTFULL</td>
<td>Optional</td>
</tr>
<tr>
<td>USRPRF</td>
<td>User profile</td>
<td>Qualifier list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier l: User profile</td>
<td>Generic name, name, *ALL</td>
<td></td>
</tr>
<tr>
<td>PCTFULL</td>
<td>Percent full</td>
<td>0.01-100.0, 99.90</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**Select by (SELECT)**

Specifies what criteria is used to select the user profiles to include in the report.

**USRPRF**

User profiles are selected for the report based on the profile name specified for the USRPRF parameter.

**PCTFULL**

User profiles are selected for the report based on the value specified for the PCTFULL parameter.
User profile (USRPRF)

If *USRPRF was specified for the Select by (SELECT) parameter, you must specify the user profiles to be included in the report.

*ALL All user profiles will be included in the report.
user-name The name of the user profile to be included in the report.
generic-user-name The generic name of the user profile to be included in the report. A generic name is a character string of one or more characters followed by an asterisk (*).

Percent full (PCTFULL)

If *PCTFULL was specified for the Select by (SELECT) parameter, you must enter a value which will be used as the percentage full. User profiles that are at least as full as the percentage specified on this parameter will be included in the report. The value specified must be between 0.01 and 100.00.

99.90 User profiles that are at least 99.9 percent filled with entries will be included in the report.

percent-full A value, ranging from 0.01 through 100.00, for the percent full selection value.

Examples

PRTPRFINT SELECT(*PCTFULL) PCTFULL(99.00)

This command prints a report of user profile internal information for all of the user profiles that are at least 99 percent full.

Error messages

*ESCAPE Messages

CPF8304 User does not have required special authorities.

CPF8307 Command &1 in use in another job.
Print Publicly Auth Objects (PRTPUBAUT)

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

The Print Publicly Authorized Objects (PRTPUBAUT) command allows you to print a report of the specified objects that do not have public authority of *EXCLUDE. For *PGM objects, only the programs that do not have public authority of *EXCLUDE that a user can call (the program is either user domain or the system security level (QSECURITY system value) is 30 or below) will be included in the report. This is a way to check for objects that every user on the system is authorized to access.

This command will print two reports. The first report (Full Report) will contain all of the specified objects that do not have public authority of *EXCLUDE. The second report (Changed Report) will contain the objects that now do not have public authority of *EXCLUDE that did have public authority of *EXCLUDE or did not exist when the PRTPUBAUT command was previously run. If the PRTPUBAUT command was not previously run for the specified objects and library or folder, there will be no 'Changed Report'. If the command has been previously run, but no additional objects do not have public authority of *EXCLUDE, then the 'Changed Report' will be printed but there will be no objects listed.

The reports will contain the following information:

- The object type specified on the command (if object type is not *DOC or *FLR).
- The name of the library specified on the command (if object type is not *BLKSF, *DIR, *DOC, *FLR, *SOCKET, *STMF, or *SYMLNK).
- The name of the folder the documents are in (if object type is *DOC).
- The name of the directory the documents are in (if object type is *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK).
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each object that does not have *PUBLIC authority of *EXCLUDE. Each entry contains the following information:
  - The name of the library the object is in (if object type is not *BLKSF, *DIR, *DOC, *FLR, *SOCKET, *STMF, or *SYMLNK).
  - The name of the folder the object is in (if object type is *FLR).
  - The name of the object.
  - The owner of the object.
  - The authorization list securing the object.
  - The special value for the *PUBLIC authority (e.g. *ALL or *CHANGE).
  - The sensitivity level of the document or folder (if object type is *DOC or *FLR).
  - An indicator for the individual authorities that *PUBLIC has to the program (‘X’ or ‘ ’) (if object type is not *DOC or *FLR).

The file QPBXXXXXXXX (where ‘XXXXXXXX’ is the object type specified on the command) in library QUSRYS contains information from the last time the PRTPUBAUT command was run. If object type is not *BLKSF, *DIR, *DOC, *FLR, *SOCKET, *STMF, or *SYMLNK there is a member within the file, with the same name as the library, for each library that has been previously specified on the command. If a special value is specified for the library name (for example, *USRLIBL), then the ‘*’ will be replaced with a ‘Q’ in the member name. For object types that don’t require a library to be specified (e.g. *USRPRF), the library name is QSYS. System file QAOBJAUT in library QSYS with format name of QSYDSAUT is the model file for the file.
If the object type is *FLR, the first member will contain the information from the previous time *FLR was specified on the command. System file QASECDLO in library QSYS with format name of QSECDLO is the model file for the file.

If the object type is *DOC, there is a member within the file for each folder that has been previously specified on the command. The member name will be the same as the system name of the folder. System file QASECDLO in library QSYS with format name of QSECDLO is the model file for the file.

If the object type is *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK, there is a member within the file for each directory that has previously been specified in the **Directory (DIR)** parameter. The member names are based on the order the directories are processed. The member naming convention is x00000001, x00000002, and so on. The first character in the member name will either be N or Y. This character indicates if the subdirectories were searched when the data was gathered. N indicates the subdirectories were not searched, Y indicates they were searched. Once a member name has been assigned to a directory, the numeric portion with the appropriate prefix is used for all of the object types listed above. The system file QASECDIR in library QSYS with format name of QSECDIR is the model file for the file.

**Note:** The file QASECGFIPB in library QUSRYS contains the file ID values of every directory that has been processed and the Nxxxxxxxxx member name that has been assigned to it. The system file QASECGFI in library QSYS with format name of QSECGFI is the model file for the file.

**Restriction:** You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

## Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CHGRPTONLY</strong></td>
<td>Changed report only</td>
<td>*NO, *YES</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td><strong>FILAUT</strong></td>
<td>Print file authority</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>CMDAUT</strong></td>
<td>Print command authority</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>PGMAUT</strong></td>
<td>Print program authority</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>JOBDAUT</strong></td>
<td>Print JOB authority</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>FLR</strong></td>
<td>Folder</td>
<td>Character value</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>DIR</strong></td>
<td>Directory</td>
<td>Path name</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>SCHSUBDIR</strong></td>
<td>Search subdirectory</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Object type (OBJTYPE)

This is a required parameter.

The type of object to search for. For a complete list of object types, press the F4 key when prompting this parameter.

*object-type*
  The type of object to be processed.

Changed report only (CHGRPTONLY)

Specifies whether just the changed report should be printed.

*NO*
  The full and changed reports will be printed.

*YES*
  Only the changed report will be printed.

Library (LIB)


The name of the library to search for objects with public authority that is not *EXCLUDE.

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

*CURLIB*
  The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched. If the ASP device (ASPDEV) parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*USRLIBL*
  If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the ASP device (ASPDEV) parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*ALL*
  All the libraries in the auxiliary storage pools (ASPs) specified for the ASP device (ASPDEV) parameter are searched.

*ALLUSR*
  All user libraries in the auxiliary storage pools (ASPs) defined by the ASP device (ASPDEV) parameter are searched.

  User libraries are all libraries with names that do not begin with the letter Q except for the following:

  #CGULIB  #DSULIB  #SEULIB
  #COBLIB  #RPGLIB
  #DFULIB  #SDALIB
Although the following libraries with names that begin with the letter Q are provided by IBM,
you typically contain user data that changes frequently. Therefore, these libraries are also 
considered user libraries:

<table>
<thead>
<tr>
<th>Name</th>
<th>Name</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>QDSNX</td>
<td>QRCLxxxxx</td>
<td>QUSRJJS</td>
</tr>
<tr>
<td>QGPL</td>
<td>QSRVAGT</td>
<td>QUSRINFO5K</td>
</tr>
<tr>
<td>QGPL3B</td>
<td>QSY52</td>
<td>QUSRNOTES</td>
</tr>
<tr>
<td>OMGTC</td>
<td>QSY52xxxxx</td>
<td>QUSROND</td>
</tr>
<tr>
<td>OMGT2C</td>
<td>QS36F</td>
<td>QUSRPOSGS</td>
</tr>
<tr>
<td>OMPGDATA</td>
<td>QUSER3B</td>
<td>QUSRPOSSA</td>
</tr>
<tr>
<td>OMQMDATA</td>
<td>QUSRADM</td>
<td>QUSRPOYSVR</td>
</tr>
<tr>
<td>OMQMPROC</td>
<td>QUSRBRM</td>
<td>QUSRDDRAS</td>
</tr>
<tr>
<td>QPFRCDATA</td>
<td>QUSRDIRCL</td>
<td>QUSRPOYS</td>
</tr>
<tr>
<td>QRC</td>
<td>QUSRDIRDB</td>
<td>QUSRVI</td>
</tr>
</tbody>
</table>

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

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

---

**Print file authority (FILAUT)**

Specifies whether the Print Publicly Authorized Objects (PRTPUBAUT) command will be run for *FILE 
objects for each of the libraries that do not have public authority of *EXCLUDE, when *LIB is specified 
for the **Object type (OBJTYPE)** parameter.

**Note:** This parameter is only used when OBJTYPE is *LIB.

- **NO** The PRTPUBAUT command will not be run for *FILE objects for each of the libraries that does 
  not have public authority of *EXCLUDE.
- **YES** The PRTPUBAUT command will be run for *FILE objects for each of the libraries that does not 
  have public authority of *EXCLUDE.

---

**Print command authority (CMDAUT)**

Specifies whether the Print Publicly Authorized Objects (PRTPUBAUT) command will be run for *CMD 
objects for each of the libraries that do not have public authority of *EXCLUDE, when *LIB is specified 
for the **Object type (OBJTYPE)** parameter.

**Note:** This parameter is only used when OBJTYPE is *LIB.

- **NO** The PRTPUBAUT command will not be run for *CMD objects for each of the libraries that does 
  not have public authority of *EXCLUDE.
- **YES** The PRTPUBAUT command will be run for *CMD objects for each of the libraries that does not 
  have public authority of *EXCLUDE.
Print program authority (PGMAUT)

Specifies whether the Print Publicly Authorized Objects (PRTPUBAUT) command will be run for *PGM objects for each of the libraries that do not have public authority of *EXCLUDE, when *LIB is specified for the Object type (OBJTYPE) parameter.

**Note:** This parameter is only used when OBJTYPE is *LIB.

*NO  The PRTPUBAUT command will not be run for *PGM objects for each of the libraries that does not have public authority of *EXCLUDE.

*YES  The PRTPUBAUT command will be run for *PGM objects for each of the libraries that does not have public authority of *EXCLUDE.

Print JOBD authority (JOBDAUT)

Specifies whether the Print Job Description Authority (PRTJOBDAUT) command will be run for each of the libraries that does not have public authority of *EXCLUDE, when *LIB is specified for the Object type (OBJTYPE) parameter. The PRTJOBDAUT command will list all of the job descriptions in the library that do not have public authority of *EXCLUDE and have a user name specified.

**Note:** This parameter is only used when OBJTYPE is *LIB.

*NO  The PRTJOBDAUT command will not be run for each of the libraries that does not have public authority of *EXCLUDE.

*YES  The PRTJOBDAUT command will be run for each of the libraries that does not have public authority of *EXCLUDE.

Folder (FLR)

This is a required parameter if *DOC is specified for the Object type (OBJTYPE) parameter.

The name of the folder to search for documents with *PUBLIC authority that is not *EXCLUDE.

`folder-name`

The name of the folder to be searched.

Directory (DIR)

This is a required parameter if *BLKSF, *DIR, *OCKET, *STMF, or *SYMLNK is specified for the Object type (OBJTYPE) parameter.

The pathname of the directory to search for objects that do not have public authority of *EXCLUDE. Only local objects in the Root, QOpenSys, and User-Defined file systems are supported.

`directory-name`

The name of the directory to be searched.
**Search subdirectory (SCHSUBDIR)**

Specifies whether to search the subdirectories for objects to be included in the public authority report.

**Note:** This parameter is only used when OBJTYPE is *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK.

*NO  The subdirectories are not searched.
*YES  The subdirectories are searched.

**Examples**

PRTPUBAUT  OBJTYPE(*FILE)  LIB(QSYS)

This command prints both full and changed reports for the file objects in the library QSYS.

**Error messages**

**ESCAPE Messages**

CPFB304  
User does not have required special authorities.

CPFB307  
Command &1 in use in another job.
Print Private Authorities (PRTPVTAUT)

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

The Print Private Authority (PRTPVTAUT) command allows you to print a report of all the private authorities for objects of a specified type in a specified library, folder or directory. The report will list all objects of the specified type and the users that are authorized to the object. This is a way to check for different sources of authority to objects.

This command will print three reports for the selected objects. The first report (Full Report) will contain all of the private authorities for each of the selected objects.

The second report (Changed Report) will contain additions/changes to the private authorities to the selected objects if the PRTPVTAUT command was previously run for the specified objects in the specified library or folder. Any new objects of the selected type, new authorities to existing objects, or changes to existing authorities to the existing objects will be listed in the 'Changed Report'. If the PRTPVTAUT command was not previously run for the specified objects in the specified library or folder, there will be no 'Changed Report'. If the command has been previously run but no changes have been made to the authorities on the objects, then the 'Changed Report' will be printed but there will be no objects listed.

The third report (Deleted Report) will contain any deletions of privately authorized users from the specified objects since the PRTPVTAUT command was previously run. Any objects that were deleted or any users that were removed as privately authorized users will be listed in the 'Deleted Report'. If the PRTPVTAUT command was not previously run, there will be no 'Deleted Report'. If the command has been previously run but no delete operations have been done to the objects, then the 'Deleted Report' will be printed but there will be no objects listed.

The reports will contain the following information:

- The object type specified on the command (if object type is not *AUTL).
- The date and time the report was last run (not shown on the Full Report).
- The name of the folder the documents or folder are in (if object type is *DOC or *FLR).
- The name of the directory the objects are in (if object type is *BLKSF, *DIR, *SOCKET, *STMF, *SYMLNK).
- The directory’s *PUBLIC authority (if object type is *BLKSF, *DIR, *SOCKET, *STMF, *SYMLNK).
- An entry for each user that has an authority to the objects in the list. Each entry contains the following information:
  - The name of the object (only shown for the first user).
  - The owner of the object (only shown for the first user).
  - The primary group of the object (only shown for the first user).
  - The name of the authorization list securing the object (only shown for the first user if object type is not *AUTL).
  - The sensitivity level of the document or folder (if object type is *DOC or *FLR, only shown for the first user).
  - The name of the user authorized to the object.
- The special value for the user’s authority to the object (e.g. *ALL or *CHANGE).
- An indicator for the individual authorities that the user has to the object (’X’ or ’’) (if object type is not *DOC or *FLR).

The file QPVXXXXXXX (where ‘XXXXXXX’ is the object type specified on the command) in library QUSRYS contains information from the last time the PRTPVTAUT command was run. If object type is not *BLKSF, *DIR, *DOC, *FLR, *SOCKET, *STMF, or *SYMLNK there is a member within the file, with the same name as the library, for each library that has been previously specified on the command. For object types that don’t require a library to be specified (e.g. *USRPRF), the library name is QSYS. System file QAOBJAUT in library QSYS with format name of QSYDSAUT is the model file for the file.

If the object type is *FLR, the first member will contain the information from the previous time *FLR was specified on the command. System file QASECDLO in library QSYS with format name of QSECDLO is the model file for the file.

If the object type is *DOC, there is a member within the file for each folder that has been previously specified on the command. The member name will be the same as the system name of the folder. System file QASECDLO in library QSYS with format name of QSECDLO is the model file for the file.

If the object type is *FILE and the AUTTYPE parameter value is *FIELD or *ALL, the Display Object Authority (DPOBJAUT) command will be run for each file that has field level authorities associated with it. For each of these files, a spooled file by the name of QPOBJAUT will be created that contains all of the field level authority data for the file. There is no changed report support available for the field level authority data on a file.

If the object type is *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK, there is a member within the file for each directory that has previously been specified in the Directory (DIR) parameter. The member names are based on the order the directories are processed. The member naming convention is x00000001, x00000002, and so on. The first character in the member name will either be N or Y. This character indicates if the subdirectories were searched when the data was gathered. N indicates the subdirectories were not searched, Y indicates they were searched. Once a member name has been assigned to a directory, the numeric portion with the appropriate prefix is used for all of the object types listed above. The system file QASECDIR in library QSYS with format name of QSECDIR is the model file for the file.

**Note:** The file QASECGFIPV in library QUSRYS contains the file ID values of every directory that has been processed and the Nxxxxxxxxx member name that has been assigned to it. The system file QASECGFI in library QSYS with format name of QSECGFI is the model file for QASECGFIPV.

**Restriction:** You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.
## Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHGRPTONLY</td>
<td>Changed report only</td>
<td>*NO, *YES</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>LIB</td>
<td>Library Name</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>AUTTYPE</td>
<td>Authority type</td>
<td>*OBJECT, *FIELD, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>FLR</td>
<td>Folder</td>
<td>Character value</td>
<td>Optional</td>
</tr>
<tr>
<td>AUTOBJ</td>
<td>Print AUTL objects</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>DIR</td>
<td>Directory Path name</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>SCHSUBDIR</td>
<td>Search subdirectory</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
</tbody>
</table>

### Object type (OBJTYPE)

This is a required parameter.

The type of object to search for. For a complete list of object types, press the F4 key when prompting this parameter.

**object-type**

The type of object to be processed.

### Changed report only (CHGRPTONLY)

Specifies whether just the changed reports should be printed.

- **NO**  The full and changed reports are printed.
- **YES** Only the changed report and the deleted reports are printed.
**Library (LIB)**


The name of the library to search for objects to be included in the private authority report.

**Authority type (AUTTYPE)**

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

- ***OBJECT**
  - Object level authority reports are generated for the specified objects.

- ***FIELD**
  - For each data base file that has field level authorities a field level authority report is generated.
  - This value is only valid if *FILE is specified for the Object type (OBJTYPE) parameter.

- ***ALL**
  - For each data base file that has field level authorities, a field level authority report is generated.
  - Also, the object level authority reports for all the files in the specified library are generated.
  - This value is only valid if *FILE is specified for the Object type (OBJTYPE) parameter.

**Folder (FLR)**

This is a required parameter if *DOC is specified for the Object type (OBJTYPE) parameter.

The name of the folder to search for documents to be included in the private authority report.

*folder-name*

  - The name of the folder to be searched.

**Print AUTL objects (AUTLOBJ)**

Specifies whether the Display Authorization List Objects (DSPAUTLOBJ) command will be run for each of the authorization lists on the system. DSPAUTLOBJ provides a list of all the objects that are secured by a specific authorization list. This parameter is only used if the object type is *AUTL. It is ignored for all other object types.

- ***NO**
  - The DSPAUTLOBJ command will not be run for each of the authorization lists on the system.

- ***YES**
  - The DSPAUTLOBJ command will be run for each of the authorization lists on the system. The output for the command will be sent to the same output queue as the authorization list report.
**Directory (DIR)**

This is a required parameter if *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK is specified for the Object type (OBJTYPE) parameter.

The name of the directory to search for objects to be included in the private authority report. Only local objects in the Root, QOpenSys, and User-Defined file systems are supported.

*directory-name*

The name of the directory to be searched.

---

**Search subdirectory (SCHSUBDIR)**

Specifies whether to search the subdirectories for objects to be included in the private authority report.

**Note:** This parameter is only used when OBJTYPE is *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK.

*NO*  The subdirectories are not searched.

*YES*  The subdirectories are searched.

---

**Examples**

PRTPVTAUT  OBJTYPE(*FILE)  LIB(PAYROLLLIB)

This command prints the full, changed, and deleted reports for all file objects in the library PAYROLLLIB.

---

**Error messages**

***ESCAPE Messages**

CPFB304  
User does not have required special authorities.

CPFB307  
Command &1 in use in another job.
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPHT (Merge TCP/IP Host Table)
Print Queue Authority (PRTQAUT)

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

The Print Queue Authority (PRTQAUT) command allows you to print a report of the output queue and job queue authority information for the objects in the specified library. This command provides a way to check the authority attributes of the output queue and job queue objects on the system.

This command will print two reports for a library. The first report (Full Report) will contain all of the output queues and job queues in the specified library. The second report (Changed Report) will contain the output queues and job queues that have been created or had the authority attributes changed since the PRTQAUT command was last run for the library. If the PRTQAUT command was not previously run for the library, there will be no ‘Changed Report’. If the command has been previously run for the library but no additional queue information is available then the ‘Changed Report’ will be printed but there will be no queues listed.

The reports will contain the following information:

- The name of the library that was specified on the command.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each output queue and job queue that exists on the system. Each entry contains the following information:
  - The name of the library the queue is in.
  - The name of the queue.
  - The object type of the queue.
  - The owner of the queue.
  - The public authority of the queue.
  - The display data value of the output queue. For job queue objects this field will be set to *NONE.
  - The operator control value of the queue.
  - The authority to check value of the queue.

The file QSECOQOLD in library QUSRYS contains information from the last time the PRTQAUT command was run for a library. There is a member within the file, with the same name as the library, for each library that has been previously specified on the command. If a special value is specified for the library name (for example, *USRLIBL), then the ‘*’ will be replaced with a ‘Q’ in the member name. System file QASECQF in library QSYS with format name of QSECOQF is the model file for the QSECOQOLD file.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
</table>

© Copyright IBM Corp. 1998, 2006 299
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHGRPTONLY</td>
<td>Changed report only</td>
<td>*NO, *YES</td>
<td>Optional, Positional 2</td>
</tr>
</tbody>
</table>

**Library (LIB)**

This is a required parameter.

The name of the library to search for output queue and job queue objects to report.

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

*CURLIB  The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched. If the ASP device (ASPDEV) parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*USRLIBL  If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the ASP device (ASPDEV) parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*ALL  All the libraries in the auxiliary storage pools (ASPs) specified for the ASP device (ASPDEV) parameter are searched.

*ALLUSR  All user libraries in the auxiliary storage pools (ASPs) defined by the ASP device (ASPDEV) parameter are searched.

User libraries are all libraries with names that do not begin with the letter Q except for the following:

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

Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

- QDSNX
- QRLxxxxxx
- QUSRRIJS
- QUSRVxRxMx
- QGPL
- QSRVAGT
- QSRINFSKR
- QGPL3B
- QSYS2
- QUSRNOTES
- QMGTC
- QSYS2xxxxxx
- QUSROND
- QMGTC2
- QSYSF
- QUSRPOS6S
- QMPGDATA
- QUSER38
- QUSRPOSSA
- QMQMDATA
- QUSRADS
- QUSRPYMSVR
- QMMPROC
- QUSRBRM
- QUSRDDS
- QPFRODATA
- QUSRDIRCL
- QUSRsys
- QRCL
- QUSR1RDB
- QUSRV1

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

**name**  Specify the name of the library to be searched.
**Changed report only (CHGRPTONLY)**

Specifies whether just the changed report should be printed.

*NO*  The full and changed reports will be printed.

*YES* Only the changed report will be printed.

**Examples**

PRTAUT LIB(QUSR_SYS)

This command prints both full and changed reports for the output queues and job queues in the library QUSR_SYS.

**Error messages**

*ESCAPE Messages*

CPFB307

Command &1 in use in another job.
Print Subsystem Description (PRTSBSDAUT)

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

The Print Subsystem Description Authority (PRTSBSDAUT) command allows you to print a report of the subsystem descriptions in a library that contain a default user in a subsystem description entry. This command provides a way to check for subsystem descriptions that allow work to be performed on your system while running under a default user profile.

This command will print two reports for a library. The first report (Full Report) will contain all of the subsystem descriptions that contain a default user in a subsystem description entry. The second report (Changed Report) will contain the subsystem descriptions that have been changed to contain a subsystem entry with a default user since the PRTSBSDAUT command was last run for the library. If the PRTSBSDAUT command was not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional subsystem descriptions contain entries with a default user, then the 'Changed Report' will be printed but there will be no subsystem descriptions listed. Changes to user profile special authorities will not cause a 'Changed Report' to be generated.

The reports will contain the following information:
- The name of the library that was specified on the command.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each subsystem description that contains a subsystem entry with a default user specified. Each entry contains the following information:
  - The name of the library the subsystem description is in.
  - The name of the subsystem description.
  - The owner of the subsystem description.
  - The name of the default user profile specified in the subsystem entry.
  - The special authorities associated with the user profile. The special authorities that are shown are all of the special authorities that would be available when the subsystem entry is used. The special authorities shown are the special authorities that the user has, plus the special authorities that the user’s group profiles have (if the user has any groups).

The file QSECSBDOLD in library QUSRSYS contains information from the last time the PRTSBSDAUT command was run for a library. There is a member within the file, with the same name as the library, for each library that has been previously specified on the command. If a special value is specified for the library name (for example, *USRLIBL), then the *' will be replaced with a 'Q' in the member name. System file QASECSBF in library QSYS with format name of QSECSBF is the model file for the QSECSBDOLD file.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.
### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHGRPTONLY</td>
<td>Changed report only</td>
<td>*NO, *YES</td>
<td>Optional, Positional 2</td>
</tr>
</tbody>
</table>

**Library (LIB)**

This is a required LIB parameter.

The name of the library to search for subsystem descriptions contain a subsystem entry with a default user profile specified.

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

*CURLIB* The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched. If the ASP device (ASPDEV) parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*USRLIBL* If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the ASP device (ASPDEV) parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*ALL* All the libraries in the auxiliary storage pools (ASPs) specified for the ASP device (ASPDEV) parameter are searched.

*ALLUSR* All user libraries in the auxiliary storage pools (ASPs) defined by the ASP device (ASPDEV) parameter are searched.

User libraries are all libraries with names that do not begin with the letter Q except for the following:

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

Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

QDSNX QRLxxxxx QUSRIJS QUSRVxRxMx
QGPL QSRVAGT QSRINFSKR
QGPL3B QSYS2 QUSRNOTES
QMTC QSYS2xxxxx QUSROND
QMTC2 Q36F QUSRPOS6S
QMPGDATA QUSER3B QUSRPOSSA
QMQMDATA QSRADSM QSRPMYSVR
QMQMROC QUSRBRM QUSRDRARS
QPFRODATA QUSRDIRCL QUSRYS
QRL QUSRDIRDB QUSRVI

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

*ALLAVL
   All libraries in all available ASPs are searched.

*ALLUSRRAVL
   All user libraries in all available ASPs are searched. Refer to *ALLUSR for a definition of user libraries.

name Specification of the name of the library to be searched.

---

**Changed report only (CHGRPTONLY)**

Specifies whether just the changed report should be printed.

*NO The full and changed reports will be printed.

*YES Only the changed report will be printed.

---

**Examples**

PRTSBSDAUT LIB(QSYS)

This command prints both full and changed reports for all subsystem descriptions in library QSYS.

---

**Error messages**

*ESCAPE Messages

CPFB307
   Command &1 in use in another job.
Print SQL Information (PRTSQLINF)

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

The Print Structured Query Language Information (PRTSQLINF) command allows you to print information about the SQL statements in a program, SQL package, service program, or job. The information includes the SQL statements, the access plans used during the running of the statement, and a list of the command parameters which are defined either during the precompile of the source member for the object or when SQL statements are run.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OBJ     | Object      | Single values: *JOB
Other values: Qualified object name | Required, Positional 1 |
| OBJTYPE | Object type | *PGM, *SQLPKG, *SRVPGM | Optional, Positional 2 |
| OBJTYPE | Library     | *LIBL, *CURLIB |

Object (OBJ)

Specifies either the name of the object for which you want SQL information printed or *JOB indicating that the job’s SQL information is to be printed. A named object can be a program, an SQL package, or a service program.

Single values

*JOB    The SQL information for the current job is to be printed. The output will only contain information for statements which have been dynamically prepared for the job. It will not contain information for SQL statements in programs, service programs, or SQL packages used by the job.

Qualifier 1: Object

name    Specify the name of the program or SQL package for which you want information printed.

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.
Object type (OBJTYPE)

Specifies the object type of the object specified for the Object (OBJ) parameter.

*PGM  The object is a program.

*SQLPKG  The object is an SQL package.

*SRVPGM  The object is a service program.

Examples

Example 1: Printing SQL Information

PRTSQLINF PAYROLL

This command will print information about the SQL statements contained in program PAYROLL.

Error messages

*ESCAPE Messages

SQL9011  Print of SQL information failed.
Print Stop Word List (PRTSWL)

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

The Print Stop Word List (PRTSWL) command is used to print the words from an IBM-supplied or user-created stop word list.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANGID</td>
<td>Language ID</td>
<td>Character value</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>TYPE</td>
<td>Stop word list type</td>
<td>*IBM, *USER</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Language ID (LANGID)

Specifies the language identifier (ID) for the stop word list.

This is a required parameter.

Stop word list type (TYPE)

Specifies the type of stop word list to print.

*IBM The stop word list is IBM-supplied.

*USER The stop word list is user-created.

Examples

PRTSWL LANGID(ENG) TYPE(*IBM)

This command prints the IBM-supplied stop word list with the language ID ENG.

Error messages

*ESCAPE Messages
CPF8725
   &1 type stop word list not supported for language.

CPF9899
   Error occurred during processing of command.
Print System Information (PRTSYSINF)

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

The Print System Information (PRTSYSINF) command prints system information that should be maintained for disaster recovery and system verification purposes. A record of the contents of your system, such as how your system is customized and what libraries it contains, is important to your upgrade success because the information helps you do the following:

- Plan you upgrade procedures
- Evaluate the success of moving information
- Perform disaster recovery, if necessary

Parameters

None

Examples

None

Error messages

Unknown
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPTHT (Merge TCP/IP Host Table)
Print System Security Attr (PRTSYSSECA)

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

The Print System Security Attributes (PRTSYSSECA) command prints a report of security related system values and network attributes to a spooled file. The report includes the system value or network attribute name, the current value, and the recommended value.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

Parameters

None

Examples

PRTSYSSECA

This command prints a report of all security-related system values and network attributes, showing the current value and the recommended value.

Error messages

*ESCAPE Messages

CPFB304

User does not have required special authorities.
Print Point-to-Point Profile (PRTTCPPTP)

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

The Print Point-to-Point TCP/IP Profile (PRTTCPPTP) command is used to print the configuration data for a point-to-point TCP/IP profile. Printer file QPTOCPPP is used to generate the spooled file. The spooled file name will be the same as the point-to-point profile name, and the spooled file user data will be ‘PRTTCPPTP’.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFGPRF</td>
<td>Configuration profile</td>
<td>Character value</td>
<td>Required, Positional 1</td>
</tr>
</tbody>
</table>

Configuration profile (CFGPRF)

Specifies the point-to-point configuration profile to be printed.

This is a required parameter.

character-value

Specify the name of a valid configuration profile.

Examples

PRTTCPPTP   CFGPRF(ANSPROFILE)

This command prints the configuration data for point-to-point profile ANSPROFILE. The spooled file name will be ANSPROFILE and the spooled file user data will be ‘PRTTCPPTP’.

Error messages

*ESCAPE Messages

TCP83F1

Point-to-point profile &1 not printed.
Print Trace Data (PRTTRC)

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

The Print Trace (PRTTRC) command formats and writes the trace records to the selected output file. The trace records were written to a set of database files by the ENDTRC (End Trace) command and PRTTRC is used to format these trace records to a spooled output file or to a database output file. If the trace records are written to a spooled output file, printer file QPSRVTRCJ is used. The user data for the spooled file will be the same as the value specified for the DTAMBR (Data member) parameter.

Restrictions:

- To use this command, you must have service (*SERVICE) special authority, or be authorized to the Service trace function of Operating System through iSeries Navigator’s Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
- You must have authority to the library and the database files within that library where the trace data is stored.
- If DLTTRC(*YES) is specified, you must have authority to the DLTTRC (Delete Trace Data) command.
- The record format of the database output file must match the record format of the IBM-supplied output file QASCTJFL.
- The ENDTRC command that produced the trace data and the PRTTRC command that processes and formats the trace data must be run on the same release of Operating System.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTAMBR</td>
<td>Data member</td>
<td>Name</td>
<td>Optional, Positional 1</td>
</tr>
<tr>
<td>DTALIB</td>
<td>Data library</td>
<td>Name, *CURLIB</td>
<td>Optional</td>
</tr>
<tr>
<td>SLTJOB</td>
<td>Select jobs</td>
<td>Single values: *ALL Other values (up to 10 repetitions): Qualified job name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Select jobs</td>
<td>Generic name, name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: User</td>
<td>Generic name, name, *ALL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 3: Number</td>
<td>000001-999999, *ALL</td>
<td></td>
</tr>
<tr>
<td>DLTTRC</td>
<td>Delete trace</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>SORT</td>
<td>Sort by</td>
<td>*TIME, *THREAD</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Output</td>
<td>*PRINT, *OUTFILE</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTFILE</td>
<td>File to receive output</td>
<td>Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: File to receive output</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>OUTMBR</td>
<td>Output member options</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Member to receive output</td>
<td>Name, *FIRST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Replace or add records</td>
<td>*ADD, *REPLACE</td>
<td></td>
</tr>
</tbody>
</table>

**Data member (DTAMBR)**

Specifies the member name for the trace data that you want to print. The member name will be the same as the trace session identifier specified on the Start Trace (STRTRC) and End Trace (ENDTRC) commands. The member name is the same for each of the physical files that contain the trace data.

This is a required parameter.

*name*  Specify the name of the database file member that contains the trace data.

**Data library (DTALIB)**

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

*CURLIB*

The trace data is printed from files in the current library for the job. If no library is specified as the current library for the job, QGPL is used.

*name*  Specify the name of the library that contains the trace data files.

**Select jobs (SLTJOB)**

Specifies which jobs to include in the trace listing. This allows the user to reduce the size of the trace listing by selecting only a subset of the jobs that were part of the trace. Up to ten qualified job names can be specified.

**Single values**

*ALL*  All jobs that were part of the trace are included.

**Qualifier 1: Select jobs**

*generic-name*

Specify the generic name of the jobs to be included in the trace listing. 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 job name specifies all jobs with job names that begin with the generic prefix.

*name*  Specify the name of the job to be included in the trace listing.

**Qualifier 2: User**

*ALL*  All jobs that match the specified job name are included.
Specify the generic user name of the jobs to be included.

Specify the name of the user of the job to be included.

Qualifier 3: Number

*ALL    All jobs that match the specified job name and user name are included.

000001-999999 Specify the job number to further qualify the job name and user name.

Delete trace (DLTTRC)

Specifies whether trace data is deleted after is has been printed.

*YES    The trace data in the database files is deleted after the print has completed.

*NO     The trace data in the database files is saved. The DLTTRC (Delete Trace) command can be used to delete the data when it is no longer needed.

Sort by (SORT)

Specifies how the trace data for each job is sorted in the specified output file.

*THREAD  The trace data for each job is sorted by thread. If a job has multiple threads, the trace data for each thread is sorted by time.

*TIME    The trace data for each job is sorted by time. If a job has multiple threads, the trace data for all threads in the job is sorted by time. This can result in the trace output for multiple threads to be intermingled.

Output (OUTPUT)

Specifies whether the output from the command is printed with the job’s spooled output or sent to a database file.

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

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

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

**Qualifier 1: File to receive output**

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

**Qualifier 2: Library**

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

*CURLIB*

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

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

**Note:** If this command creates the file, the text is "OUTFILE created by PRTTRC command" and the public authority is *EXCLUDE.*

---

**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: Print and Delete Trace**

PRTTRC DTAMBR(TRACE8) DTALIB(TRCLIB1) DLTTRC(*YES)
This command formats and prints the trace data contained in database file members named TRACE8 in library TRCLIB1. The trace data members are removed after the trace data spooled file has been written. All jobs which were part of the trace will be part of the trace listing.

**Example 2: Print Subset Trace**

```
PRTTRC  DTAMBR(T123456789)  DTALIB(QGPL)
       SLTJOB(*ALL/QSYS/QCMN*)  DLTTRC(*YES)
```

This command formats and prints the trace data contained in database file members named T123456789 in library QGPL. The trace data members are removed after the trace data spooled file has been written. Only those traced jobs that were started under user profile QSYS and had job names that started with "QCMN" will be part of the trace listing.

**Example 3: Print Trace and Sort by Time**

```
PRTTRC  DTAMBR(MYTRACE)  DTALIB(MYTRCLIB)
       DLTTRC(*YES)  SORT(*TIME)
```

This command formats and prints the trace data contained in database file members named MYTRACE in library MYTRCLIB. The trace data members are removed after the trace data spooled file has been written. The trace records are sorted by the time the record was collected. If the traced jobs were multithreaded, the trace output is sorted by job, with all threads in that job sorted by time. The resulting output may have trace information for multiple threads intermingled.

**Example 4: Print Trace to an Output File**

```
PRTTRC  DTAMBR(BIGTRACE)  DTALIB(TRACELIB)  DLTTRC(*YES)
       OUTPUT(*OUTFILE)  OUTFILE(MYLIB/MYFILE)
```

This command stores the trace data contained in database file members named BIGTRACE in library TRACELIB into a database output file named MYFILE in library MYLIB. The trace data members named BIGTRACE are removed after the trace data has been written to the database output file.

---

**Error messages**

***ESCAPE Messages**

**CPF39CD**

Error occurred during processing of the PRTTRC command.

**CPF98A2**

Not authorized to &1 command.
The Print Trigger Program (PRTTRGPGM) command lists the programs which have been defined as a trigger program for the files in the specified library.

This command will print two reports for a library. The first report (Full Report) will contain all of the trigger programs associated with files in the specified library. The second report (Changed Report) will contain the trigger programs that now appear in the specified library and were not in the library when the PRTTRGPGM command was previously run for the library. If the PRTTRGPGM command was not previously run for the library but no additional trigger programs are in the specified library, the ‘Changed Report’ will be printed but there will be no objects listed. Changing the trigger time, trigger event or trigger update condition for a trigger program will not cause a ‘Changed Report’ to be generated.

The file QSECTRGOLD in library QUSRSYS contains information from the last time the PRTTRGPGM command was run for a library. There is a member within the file, with the same name as the library specified, for each library previously specified on the command. System file QAFDTRG in library QSYS with format name of QWHFDTRG is the model file for the QSECTRGOLD file.

**Restriction:** You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHGRPTONLY</td>
<td>Changed report only</td>
<td>*NO, *YES</td>
<td>Optional, Positional 2</td>
</tr>
</tbody>
</table>

### Library (LIB)

This is a required parameter.

The name of the library to search for files that have trigger programs.

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

*CURLIB  The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched. If the ASP device (ASPDENV) parameter is specified when this value is used, ASPDEV(*) is the only valid value.
If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the ASP device (ASPDEV) parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*ALL  All the libraries in the auxiliary storage pools (ASPs) specified for the ASP device (ASPDEV) parameter are searched.

*ALLUSR
All user libraries in the auxiliary storage pools (ASPs) defined by the ASP device (ASPDEV) parameter are searched.

User libraries are all libraries with names that do not begin with the letter Q except for the following:

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

Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

QDSNX  QRLxxxxx  QUSRIJS  QUSRVxRxMx
QGPL  QSRVAGT  QUSRINFSKR
QGPL38  QSYS2  QUSRNOTES
QMGTC  QSYS2xxxxx  QUSROND
QMGTC2  Q36F  QUSRPOSGS
QMPGDATA  QUSER3B  QUSRPOSSA
QMQUAL DATA  QUSRADM  QUSRPMYSVR
QMQUALPROC  QUSRBM  QUSRBRDARS
QPFRODATA  QUSRDIRCL  QUSRYS
QRCL  QUSRDIRDB  QUSRVI

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

Specify the name of the library to be searched.

Changed report only (CHGRPTONLY)
Specifies whether just the changed report should be printed.

*NO  The full and changed reports will be printed.
*YES  Only the changed report will be printed.

Examples

PRTTRGPGM  LIB(*ALL)

This command searches all files in all libraries and prints both full and changed trigger program reports.
Error messages

*ESCAPE Messages

CPF304

User does not have required special authorities.
Print User Objects (PRTUSROBJ)

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

The Print User Objects (PRTUSROBJ) command allows you to print a report of the objects in a library that are not created by IBM. Objects are included in the report if the "Created by user" attribute is not *IBM or QLPINSTALL. Use this command to check for user created objects that are in libraries intended for use only by IBM. For example, you may want to run this program for library QSYS to determine if it contains any non-IBM (user) objects.

Note: Some objects created by IBM will still appear in this report. For example, objects created by a PTF exit program will be included in this report. Objects are excluded from the report only when their "Created by user" attribute is either "*IBM" or "QLPINSTALL."

This command will print two reports for a library. The first report (Full Report) will contain all of the objects that have not been created by IBM. The second report (Changed Report) will contain the objects that now appear in the specified library and were not in the library when the PRTUSROBJ command was previously run for the library. If the PRTUSROBJ command was not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional objects have been added to the library that were not created by IBM, then the 'Changed Report' will be printed but there will be no objects listed.

The file QSECPUOOLD in library QUSR SYS contains information from the last time the PRTUSROBJ command was run for a library. There is a member within the file, with the same name as the library specified, for each library previously specified on the command. System file QADSPOBJ in library QSYS with format name of QLIDOBJD is the model file for the QSECPUOOLD file.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIB</td>
<td>Library</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>CHGRPTONLY</td>
<td>Changed report only</td>
<td>*NO, *YES</td>
<td>Optional, Positional 2</td>
</tr>
</tbody>
</table>

Library (LIB)

This is a required parameter.

The name of the library to search for objects that were not created by IBM.
**Changed report only (CHGRPTONLY)**

Specifies whether just the changed report should be printed.

*NO*   The full and changed reports will be printed.

*YES*  Only the changed report will be printed.

---

**Examples**

PRTUSROBJ LIB(QSYS) CHGONLY(*NO)

This command searches library QSYS for any objects that were not created by IBM and prints both full and changed reports.

---

**Error messages**

*ESCAPE Messages*

CPFB304

User does not have required special authorities.
Print User Profile (PRTUSRPRF)

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

The Print User Profile (PRTUSRPRF) command allows you to print a report containing information for the user profiles on the system. Four different reports can be printed. One contains authority type information, one contains environment type information, one contains password type information, and one contains password level type information.

The authority information report will contain the following information:
- The type of report.
- The criteria used to select user profiles in the report.
- The special authorities being selected on (if ’Select by’ is *SPCAUT).
- The user classes being selected on (if ’Select by’ is *USRCLS).
- An entry for each user profile that was selected. Each entry contains the following information:
  - The name of the user profile.
  - The names of the user’s group profiles.
    If the user profile does not have any group profiles, this field will contain ’*NONE’. If the user has group profiles, an additional entry will follow the user profile entry for each of the user’s groups. The entry will contain the group profile name and the special authorities that the group profile has.
  - An indicator for the special authorities that the user profile has (’X’ or ’ ’).
  - The user class for the user profile.
  - Whether the user profile or its group profile own objects created by this user profile.
  - What authority is given to the user’s group profile for newly created objects (if the owner value is *USRPRF).
  - The limited capability value for the user profile.

The environment information report will contain the following information:
- The type of report.
- The criteria used to select user profiles in the report.
- The special authorities being selected on (if ’Select by’ is *SPCAUT).
- The user classes being selected on (if ’Select by’ is *USRCLS).
- An entry for each user profile that was selected. Each entry contains the following information:
  - The name of the user profile.
  - The name of the user’s current library.
  - The name of the user’s initial menu, and the library it is in.
  - The name of the user’s initial program, and the library it is in.
  - The name of the user’s job description, and the library it is in.
  - The name of the user’s message queue, and the library it is in.
  - The name of the user’s attention program, and the library it is in.

The password information report will contain the following information:
- The type of report.
- The criteria used to select user profiles in the report.
• The special authorities being selected on (if ‘Select by’ is *SPCAUT).
• The user classes being selected on (if ‘Select by’ is *USRCLS).
• The value of the QPWDEXPTV system value (for reference if the user’s password expiration interval is *SYSVAL).

An entry for each user profile that was selected. Each entry contains the following information:
– The name of the user profile.
– The status of the user profile.
– The number of sign-on attempts that were not valid.
– The ‘no password’ indicator (‘X’ if the user doesn’t have a password, ‘ ’ if it does).
– Whether the password is managed locally.
– The user’s previous sign-on date.
– The date the user’s password was last changed.
– The user’s password expiration interval.
– Whether the user’s password is set to expired.

The password level information report will contain the following information that can be used to determine if the system is ready to change password levels.
• The type of report.
• The criteria used to select user profiles in the report.
• The special authorities being selected on (if ‘Select by’ is *SPCAUT).
• The user classes being selected on (if ‘Select by’ is *USRCLS).

An entry for each user profile that was selected. Each entry contains the following information:
– The name of the user profile.
– The ‘password present for level 0 or 1’ indicator (*YES if the user has a password, *NO if the user does not have a password, or *UNKNOWN if the password information was not available).
– The ‘password present for level 2 or 3’ indicator (*YES, *NO, or *UNKNOWN).
– The ‘password present for NetServer’ indicator for Windows 95 and 98 NetServer passwords (*YES, *NO, or *UNKNOWN).

Note: The Display Security Attributes (DSPSECA) command can be used to display the current and pending password level for the system. The password level can be changed by changing the QPWDLVL system value.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>Type of information</td>
<td>*ALL, *AUTINFO, *ENVINFO, *PWDINFO, *PWDLVL</td>
<td>Optional</td>
</tr>
<tr>
<td>SELECT</td>
<td>Select by</td>
<td>*SPCAUT, *USRCLS, *MISMATCH</td>
<td>Optional</td>
</tr>
</tbody>
</table>
**Type of information (TYPE)**

The type of information that can be printed for the selected user profiles.

* **ALL**  
  All of the reports are printed for the selected user profiles.

* **AUTINFO**  
  A report containing the authority type information for the selected user profiles is printed.

* **ENVINFO**  
  A report containing the environment type information for the selected user profiles is printed.

* **PWDINFO**  
  A report containing the password type information for the selected user profiles is printed.

* **PWDLVL**  
  A report containing the password level type information for the selected user profiles is printed.

This report can be used to determine which user profiles have passwords that are used at the different password levels.

**Select by (SELECT)**

Specifies what criteria is used to select the user profiles to include in the report.

* **SPCAUT**  
  User profiles will be selected for the report based on special authorities.

* **USRCLS**  
  User profiles will be selected for the report based on user class.

* **MISMATCH**  
  User profiles will be selected for the report based on their special authorities not being the default values assigned to their user class.

  **Note**: The defaulted special authorities for user classes changed in V3R7. Therefore, when running this report for profiles created prior to V3R7, you may notice a larger than expected number of profiles that do not match the default values.

**Special authorities (SPCAUT)**

If *SPCAUT was specified for the Select by (SELECT) parameter, it specifies which special authorities should be used to select users. User profiles with any of the special authorities specified for this parameter will be included in the report. A maximum of 9 special authorities can be specified.

You can enter multiple values for this parameter.

* **ALL**  
  All user profiles will be included in the report.

  Or select one or more of the following values (9 maximum):

* **ALLOBJ**  
  User profiles with *ALLOBJ special authority will be included in the report.
**AUDIT**
User profiles with *AUDIT special authority will be included in the report.

**IOSYSCFG**
User profiles with *IOSYSCFG special authority will be included in the report.

**JOBCTL**
User profiles with *JOBCTL special authority will be included in the report.

**SAVSYS**
User profiles with *SAVSYS special authority will be included in the report.

**SECADM**
User profiles with *SECADM special authority will be included in the report.

**SERVICE**
User profiles with *SERVICE special authority will be included in the report.

**SPLCTL**
User profiles with *SPLCTL special authority will be included in the report.

**NONE**
User profiles with no special authorities will be included in the report.

---

**User class (USRCLS)**

If *USRCLS was specified for the Select by (SELECT) parameter, it specifies that user classes should be used to select users. User profiles with a user class that is specified for this parameter will be included in the report. A maximum of 5 user classes can be specified.

You can enter multiple values for this parameter.

**ALL**  
All user profiles will be included in the report.

Or select one or more of the following values (5 maximum):

**USER**
User profiles with *USER user class will be included in the report.

**SYSOPR**
User profiles with *SYSOPR user class will be included in the report.

**PGMR**
User profiles with *PGMR user class will be included in the report.

**SECADM**
User profiles with *SECADM user class will be included in the report.

**SECOFR**
User profiles with *SECOFR user class will be included in the report.

---

**Examples**

```
PRTUSRPRF TYPE(*ALL) SELECT(*SPCAUT) SPCAUT(*ALLOBJ *SECADM)
```

This command prints all four reports for user profiles that have either *ALLOBJ or *SECADM special authority.
Error messages

*ESCAPE Messages

CPF8304
   User does not have required special authorities.

CPF8307
   Command &1 in use in another job.
The Power Down System (PWRDWNSYS) command prepares the system for ending and then starts the power-down sequence. All active subsystems are notified that the system is being powered down; no new jobs or routing steps can be started by any subsystem. For example, jobs that are on a job queue will not be started. In addition, jobs that are on a job queue as a result of a Transfer Job (TFRJOB) command will not complete. They are removed from the job queue during the subsequent initial program load (IPL), and their job logs are produced.

When the system is powered down with the *CNTRLD option, a vary off of configuration objects is initiated, but may not complete before the power down completes. When the system is powered down with the *IMMED option, no vary off of configuration objects is performed.

Notes:
1. If network server descriptions are configured on the system, all NWSDs should be varied off before the PWRDWNSYS command is issued to ensure the integrity of system and user data associated with each network server.
2. If tape units are installed on the system, all tape reels that are on the device(s) should be unloaded before the system is powered down to ensure the integrity of data on the tapes.
3. If the system has a primary partition, powering down the primary partition will cause the other partitions to power down. Ensure the other partitions are ready to be powered down before powering down the primary partition.
4. If independent auxiliary storage pool (ASP) devices are configured on the system, all independent ASPs should be varied off before the PWRDWNSYS command is issued to ensure the integrity of data associated with each independent ASP.
5. The registered Prepower down system exit point (QIBM_QWC_PWRDWNSYS) has two possible formats. Format PWRD0100 can be used to add a program that is called when the PWRDWNSYS command is used. Format PWRD0200 can be used to add one or more programs that are called when the PWRDWNSYS command is used. These exit programs can perform clean up functions before the system is powered down.
6. When changing the QENDJOBLMT and QPWRDWNLMT system values, specify values so that QPWRDWNLMT is greater than QENDJOBLMT. The values need to allow enough time for system-supplied end-of-job functions such as completing commitment control processing and closing database files.

Restrictions:
1. To run this command, the user must have job control (*JOBCTL) authority.
2. When *IMGCLG is specified on the IPL source (IPLSRC) parameter you need the following authorities:
   - execute (*EXECUTE) authority to library QUSRYS
   - use (*USE) authority to the image catalog specified by the Image catalog (IMGCLG) parameter
   - use (*USE) authority to the virtual device description
   - execute (*X) authority to each directory in the image catalog path name
3. If you have a user exit program defined to the Prepower down system exit point (QIBM_QWC_PWRDWNSYS) for format PWRD0100, then the user must have use (*USE) authority to
the user exit program and execute (*EXECUTE) authority to the library that contains that program. If not, then this user exit program will not be called and the system will continue to power down.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTION</td>
<td>How to end</td>
<td>*CNTRLD, *IMMED</td>
<td>Optional, Positional 1</td>
</tr>
<tr>
<td>DELAY</td>
<td>Controlled end delay time</td>
<td>1-99999, 3600, *NOLIMIT</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>RESTART</td>
<td>Restart options</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Restart after</td>
<td>*NO, *YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>power down</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Restart type</td>
<td>*IPLA, *SYS, *FULL</td>
<td></td>
</tr>
<tr>
<td>IPLSRC</td>
<td>IPL source</td>
<td>*PANEL, A, B, D, *IMGCLG</td>
<td>Optional</td>
</tr>
<tr>
<td>IMGCLG</td>
<td>Image catalog</td>
<td>Name</td>
<td>Optional</td>
</tr>
<tr>
<td>ENDSBSOPT</td>
<td>End subsystem option</td>
<td>Single values: *DFT</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Other values (up to 3 repetitions): *NOJOBLOG, *CHGPTY, *CHGTSL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMEOUTOPT</td>
<td>Timeout option</td>
<td>*CONTINUE, *MSD, *SYSREFCDE</td>
<td>Optional</td>
</tr>
<tr>
<td>CONFIRM</td>
<td>Confirm</td>
<td>*ENVVAR, *INTERACT, *YES, *NO</td>
<td>Optional</td>
</tr>
</tbody>
</table>

### How to end (OPTION)

Specifies whether the system allows the active subsystem to end processing of active jobs in a controlled manner (which lets the application program perform end processing), or whether the system ends the jobs immediately. In either case, the system does perform certain job-cleanup functions.

***CNTRLD**

The subsystem, in the time specified by the Controlled end delay time (DELAY) parameter ends all active jobs in a controlled manner. During that time, programs running in those jobs are allowed to perform cleanup (end-of-job processing). When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job. If the possibility exists that an active job could begin to loop or send an inquiry message to QSYSOPR, you should specify a time delay using the DELAY parameter.

***IMMED**

The subsystem ends all active jobs immediately. When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job and the QENDJOBLMT system value specifies a time limit. Other than by handling the SIGTERM signal, the programs running in those jobs are not allowed to perform any cleanup. A minimum amount of time is required when *IMMED is specified. The amount of time allowed for cleanup when *IMMED is specified is controlled by the system values QENDJOBLMT and QPWDRWNLMT.

**Note:** The *IMMED value might cause undesirable results if data has been partially updated. This value should be used only after a controlled end has been attempted unsuccessfully.
**Note:** When *IMMED* is specified while the system is operating under auxiliary power, or if the delay time specified in the **Controlled end delay time (DELAY)** parameter ends while the system is under auxiliary power, the system is immediately powered-down without additional job cleanup activity.

---

### Controlled end delay time (DELAY)

Specifies the amount of time, in seconds, that the system allows a controlled end to be performed by the active subsystems. If the end of job routines are not finished in the specified delay time, any remaining jobs are ended immediately.

- **3600** The amount of time in which to complete a controlled end of processing is limited to 3600 seconds.

- **NOLIMIT** The system does not power down until the last job is complete.

  **Note:** If *NOLIMIT* is specified, a batch job could begin to loop, and the system does not power down.

- **1-99999** Specify the maximum amount of delay time, in seconds, in which a controlled end can be performed.

---

### Restart after power down (RESTART)

Specifies whether the system ends and powers down, or whether the system ends and then starts again in unattended mode.

The second element of this parameter specifies the point from which the initial program load (IPL) restarts. Specifying *SYS* rather than *FULL* can reduce the time required to restart the system.

**Element 1: Restart after power down**

- **NO** The system ends and powers down.

- **YES** If the system is on utility power, it undergoes end of processing (but does not power down) and then does an abbreviated IPL. If the system is on auxiliary power, it powers down and an automatic-IPL occurs when utility power is restored (if QPWRSTIPL system value is set to ‘1’). When the system starts again or when an automatic-IPL occurs, the IPL proceeds in an unattended mode. In unattended mode, displays such as the IPL options display are not shown.

**Element 2: Restart type**

- **IPLA** The value specified on the Change IPL Attributes (CHGIPLA) command is used. To determine the current setting for this value, use the Display IPL Attributes (DSPIPLA) command.

- **SYS** The operating system is restarted. The hardware will only be restarted when required by the system.

- **FULL** All portions of the system, including the hardware, are restarted.
IPL source (IPLSRC)

Specifies whether an initial-program-load (IPL) is started from the A-source, B-source or D-source of the system. This parameter allows you to control which Licensed Internal Code (LIC) storage source of the system to IPL. Also, the source of the system determines where LIC program temporary fixes (PTFs) are applied. This parameter also allows the system to be upgraded to a new release from an install image on DASD.

Source Considerations

LIC has three storage areas known as the A-source, the B-source and the D-source. The D-source is the install media. The A- and B-sources are part of the system memory. Initially, the A- and B-sources are identical, but when Licensed Internal Code fixes are performed temporarily (PTF), the temporary fixes are stored on the B-source. When the same fixes become permanent, they are copied from the B-source to the A-source; therefore, the fixes reside on both the A-source and the B-source.

When you want to send temporary fixes to the B-source, you must start the system from the A-source, which causes the fixes to be sent to the B-source.

When you start the system from the A-source, you are running the system from the permanent fixes. When you start the system from the B-source, you are running the system from a mixture of temporary and permanent fixes. When you start the system from the D-source, you are using the Licensed Internal Code loaded from the install media.

It is recommended that you specify RESTART(*YES); otherwise, you cannot be assured which source of the system is actually started. This precaution can save you some time.

*PANEL

The system is started from the source that is currently shown on the operator’s panel, the A-source, the B-source, or the D-source.

A The system is started from the A-source.
B The system is started from the B-source.
D The system is started from the D-source, the install media.

*IMGCLG

The system is started from the image catalog specified with the Image catalog (IMGCLG) parameter. RESTART(*YES) must be used when this option is selected.

Image catalog (IMGCLG)

Specifies the image catalog used when IPLSRC(*IMGCLG) is selected. After the system is powered down, an install using the specified image catalog is performed. See the Work with Catalog Entries (WRKIMGCLGE) command for more information. RESTART(*YES) must be used when this parameter is specified.

name Specify the name of the image catalog in library QUSRSYS.
End subsystem option (ENDSBSOPT)

Specifies the options to take when ending the active subsystems. In general, specifying these options will improve the performance of the PWRDWSYS command. Each option has certain side effects that you need to analyze before using that option.

This parameter has no effect on jobs that are already in the ending status.

*DFT  The subsystems will end with no special ending options.
    - Joblogs will be produced.
    - The run priority will not change.
    - The timeslice value will not change.

*NOJOBLOG
No joblogs will be created for jobs that are ended due to this command being invoked. This includes subsystem monitor jobs and all user jobs in the subsystem. This option can significantly reduce the amount of time necessary to complete the PWRDWSYS command. However, if a problem occurs in a job, there will be no joblog to record the problem, which may make problem diagnosis difficult or impossible.

Note: If OPTION(*IMMED) is specified, then no joblogs are produced during PWRDWSYS regardless of the End subsystem option (ENDSBSOPT) parameter. However, these joblogs will still be produced on the next IPL of the system unless the *NOJOBLOG option is specified. Therefore, if you specify OPTION(*IMMED) ENDSBSOPT(*NOJOBLOG), the system will not power down more quickly, but the subsequent IPL may be faster.

*CHGPTY
The CPU priority of jobs that are ending is changed to a higher value (worse priority). The remaining active jobs on the system may have better performance when *CHGPTY is specified. However, jobs that are ending may take longer to finish. This option is ignored if the system is ending controlled. But if the DELAY time limit expires, this option will take effect immediately.

*CHGTSL
The timeslice of jobs that are ending is changed to a lower value. The remaining active jobs on the system may have better performance when *CHGTSL is specified. However, jobs that are ending may take longer to finish. This option is ignored if the system is ending controlled. But if the DELAY time limit expires, this option will take effect immediately.

Timeout option (TIMOUTOPT)

Specifies the option to take when the system does not end within the time limit specified by the QPWRDWNLMT system value. If this time limit is exceeded, the subsequent IPL will be abnormal regardless of the value specified for this parameter.

*CONTINUE
The system will ignore the timeout condition and continue powering the system down. If RESTART(*YES) is specified, the system will restart automatically. A minimum of information will be available for service to debug the system.

*MSD
The system will issue a main store dump which can be used by service to debug the system. If the main store dump manager is configured correctly, the system will restart after the dump is finished.

*SYSREFCDE
The system will display system reference code B900 3F10 and the system will stop. This will allow service to debug the system.
Confirm (CONFIRM)

Specifies whether the request should be confirmed before the system is powered down.

*ENVVAR

The value in environment variable QIBM_PWRDWNSYS_CONFIRM is used to determine whether the request should be confirmed. If the value is set to *INTERACT, *YES, or *NO, the action described below for that value is taken. If the environment variable is not defined or not set to one of these values, then there is no confirmation. System initiated power downs do not use the environment variable.

*INTERACT

A confirmation panel is displayed when the PWRDWNSYS command is issued in an interactive job. There is no confirmation when the PWRDWNSYS command is issued in a non-interactive job.

*YES

A confirmation panel is displayed when the PWRDWNSYS command is issued in an interactive job. An inquiry message is sent to QSYSOPR when the PWRDWNSYS command is issued in a non-interactive job.

*NO

There is no confirmation when the PWRDWNSYS command is issued.

Examples

Example 1: Performing An Immediate End

PWRDWNSYS OPTION(*IMMED)

This command causes the system to perform an immediate end without allowing any active jobs to perform cleanup routines. Once the system completes its end functions, it starts the power-down sequence.

Example 2: Specifying a Controlled End

SBMJOB JOB(LASTJOB) JOBD(QBATCH) JOBPRTY(9) JOBQ(QBATCH)
RQSDTA('PWRDWNSYS *CNTRLD 3600')

This command submits a low priority batch job that, when run, causes the system to perform a controlled end. The controlled end is allowed one hour (3600 seconds) for completion before any remaining jobs are ended. This method of issuing the PWRDWNSYS command could be used to allow other higher priority jobs on job queue QBATCH (including those that are on the queue as a result of the Transfer Job (TFRJOB) command) to be completed before the PWRDWNSYS command is run. There must be an active subsystem for which the QBATCH job queue is a source of work.

Example 3: Specifying a Controlled End With No Time Limit

PWRDWNSYS OPTION(*CNTRLD) RESTART(*YES)

This command causes the system to perform a controlled end with no time limit. When all jobs in the system have completed, the system prepares for ending and starts an IPL.

After PWRDWNSYS OPTION(*CNTRLD) is entered, and before the delay time ends, this command can be overridden by entering PWRDWNSYS OPTION(*IMMED). In this case, the values specified or defaulted for the RESTART parameter on the second command also override the values specified or defaulted for the first command.
Example 4: Changing the IPL Source After Immediate End

```
PWRDWN SYS  OPTION(*IMMED)  RESTART(*YES)  IPLSRC(A)
```

This command causes the system to end immediately and change the IPL source to A. When the system restarts, it IPLs on the A source.

Example 5: Allowing the Operating System to Determine the Restart Point

```
PWRDWN SYS  OPTION(*IMMED)  RESTART((*YES *SYS))
```

This command causes the IPL to restart at the point determined by the operating system.

Example 6: Changing the Time Out Option.

```
PWRDWN SYS  OPTION(*IMMED)  TIMOUTOPT(*MSD)
```

This command causes the system to end immediately. If the QPWRDWNLMT system value is exceeded, the system will dump the main storage. If the main store dump manager is configured correctly, the system will restart. Otherwise, the B900 3F10 system reference code will be displayed and the system will halt.

Example 7: Installing a New Release of the Operating System.

```
PWRDWN SYS  RESTART(*YES)  IPLSRC(*IMGCLG)  IMGCLG(MYCAT1)
```

This command causes the system to end and then start installing a new release of the operating system from the image catalog MYCAT1.

---

**Error messages**

*ESCAPE Messages*

- CPF1001
  - Wait time expired for system response.

- CPF1036
  - System powering down with *CNTRLD option.

- CPF1037
  - System powering down with *IMMED option.

- CPF1038
  - No authority to use command.

- CPF1091
  - Function check occurred in system arbiter.

- CPF18C7
  - PWRDWN SYS not allowed to continue.

- CPFBC42
  - Verification for image catalog &1 failed.
The Query Document Library (QRYDOCLIB) command allows you to search for documents within the document library. A document list is created containing the results of the search. You can also copy information about the documents that satisfy the search request into a database file for processing.

When the QRYDOCLIB command is run, a document list object is created. The document list object is created regardless of whether an output file is produced unless the user specifies *NONE for the DOCL parameter. This document list object is used by the OfficeVision/400 product as well as the SAVDLO command.

**Restrictions:**
- The current user of this command must have the authority to work on behalf of the specified user ID address. To work on behalf of other users, the user must have special permission granted with the Grant User Permission (GRTUSRPMN) command.
- The format of the output file must be the same as OSIQDL of the system file, QSYS/QAOSIQDL.
- If several QRYDOCLIB commands are run at the same time, the document list name (Document list (DOCL) parameter) and the output file name or member name (File to receive output (OUTFILE) parameter) must be different for each of these QRYDOCLIB commands.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>QRYDFN</td>
<td>Query definition</td>
<td>Single values: *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Start query</td>
<td>*IF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Search criteria</td>
<td>Values (up to 49 repetitions): Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Profile</td>
<td>*ACTDATE, *ALWRPL, *ASP, *AUTHOR, *CHGDATE,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>*CMPDATE, *CPYLST, *CRTDATE, *DOCCLS, *DOCD,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>*DOCDATE, *DOCTYPE, *EXPDATE, *FILDATE,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>*IDXDATE, *KWD, *OWNER, *PROJECT, *REF,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>*REVDATE, *STATUS, *SUBJECT, *USEDATE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 3: Compare value</td>
<td>Character value, *YES, *NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 4: Logical operator</td>
<td>*AND, *OR</td>
<td></td>
</tr>
<tr>
<td>FLR</td>
<td>In folder</td>
<td>Single values: *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Other values (up to 100 repetitions): Character value, *NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCHSUBFLR</td>
<td>Search subfolder</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>QRYTXT</td>
<td>Query text</td>
<td>Single values: *NONE Other values: Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Start query</td>
<td>*IF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Search criteria</td>
<td>Values (up to 30 repetitions): Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Phrase</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Type of matching</td>
<td>*ALL, *EXACT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 3: Allow synonyms</td>
<td>*NO, *YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 4: Logical operator</td>
<td>*OR, *AND, *ANDNOT</td>
<td></td>
</tr>
<tr>
<td>TXTLANGID</td>
<td>Language ID</td>
<td>Character value, *JOB</td>
<td>Optional</td>
</tr>
<tr>
<td>DOCI</td>
<td>Document list</td>
<td>Character value, *DFI, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>TEXT</td>
<td>Text</td>
<td>Character value, *BLANK</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTFILE</td>
<td>File to receive output</td>
<td>Single values: *NONE Other values: Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: File to receive output</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>OUTMBR</td>
<td>Output member options</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Member to receive output</td>
<td>Name, *FIRST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Replace or add records</td>
<td>*REPLACE, *ADD</td>
<td></td>
</tr>
<tr>
<td>OUTDTATYP</td>
<td>Type of data for output</td>
<td>Single values: *DFI, *ALL Other values (up to 21 repetitions): ACTDATE, AUTHOR, CHGDATE, CMPDATE, CPYLST, CRTDATE, DOCCLS, DOCD, DOCDATE, EXPDATE, FILCAB, FILDATE, *IDP, IDXDATE, KWD, PROJECT, REF, REVDATE, STATUS, SUBJECT, USEDATE</td>
<td>Optional</td>
</tr>
<tr>
<td>USRID</td>
<td>User identifier</td>
<td>Single values: *CURRENT Other values: Element list</td>
<td>Optional, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Element 1: User ID</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Address</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td>TMLMT</td>
<td>Time limit</td>
<td>1-9999, *NOMAX</td>
<td>Optional</td>
</tr>
<tr>
<td>SELMT</td>
<td>Selection limit</td>
<td>1-32767, *NOMAX</td>
<td>Optional</td>
</tr>
<tr>
<td>ORDER</td>
<td>Order by</td>
<td>Single values: *NONE Other values (up to 5 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Profile</td>
<td>*ACTDATE, *ASIP, *AUTHOR, CHGDATE, CMPDATE, CPYLST, CRTDATE, DOCCLS, DOCD, DOCDATE, DOCTYPE, EXPDATE, FILDATE, IDXDATE, KWD, OWNER, PROJECT, REF, REVDATE, STATUS, SUBJECT, USEDATE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Selection order</td>
<td>*ASCEND, DESCEND</td>
<td></td>
</tr>
<tr>
<td>CMDCHRID</td>
<td>Command character identifier</td>
<td>Single values: *SYSVAL, *DEVD Other values: Element List</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Graphic character set</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Code page</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>
Query definition (QRYDFN)

Specifies what query definition selection values are used to select the documents. The values you specify on this parameter are used to search the document library. If values other than *NONE are specified on both the Query text (QRYTXT) parameter and this parameter, only documents that match both sets of values are selected. If you specify *NONE on both parameters, all documents to which you are authorized are selected.

Single values

*NONE

No query definition is used to select the documents.

Element 1: Start query

*IF A query definition is used to select the documents.

Element 2: Search criteria

To specify the conditions under which documents are selected, a set of values is specified for each condition. Each set must contain exactly four values:

1. The name of the document profile parameter to be compared (from the list that follows)
2. One of the relational operator values (from the list that follows)
3. The compare value
4. One of the logical operators *AND, *OR, or *ANDNOT

Values 1 and 3 are compared for the relationship specified by value 2. Each QRYDFN relational set must be enclosed in parentheses. A maximum of 49 sets of values can be specified.

Element 1: Profile

Specifies the name of the document profile parameter to be compared.

*ACTDATE Action due date

*ALWRPL Allow document replacement

*ASP Auxiliary storage pool ID

*AUTHOR Document author

*CHGDATE Last change date

*CMPDATE Completion date

*CPYLST Copy list

*CRTDATE Create date

*DOCCLS Document class
Element 2: Relational operator

The relational operator indicates the relationship that must exist between the profile parameter contents in the document and the value specified as the compare value parameter of the query definition for the relationship to be true.

The *CT operator is used to perform a context search. It asks the system to determine whether the character string specified by the compare value is contained anywhere in the profile parameter.

The *BG operator is used to perform a search that compares the compare value with the beginning of the profile parameter. The profile parameter is truncated or extended as necessary to match the length of the specified value. It asks the system to determine whether the character string specified by the value is contained at the beginning of the profile parameter.

Some operators are not allowed for some profile parameters. In this case, a diagnostic message followed by an escape message is sent.

The following cases are not valid:
• The *ALWRPL (allow document replacement) is a YES/NO switch. The *EQ operator is the only operator allowed with *ALWRPL.
The *CT and *BG operators are not allowed with the *ASP value or date values such as *CRTDATE and *EXPDATE.

- *EQ  Equal
- *GT  Greater than
- *LT  Less than
- *NE  Not equal
- *GE  Greater than or equal
- *NL  Not less than
- *LE  Less than or equal
- *NG  Not greater than
- *CT  Contains
- *BG  Begins

Element 3: Compare value

Specifies the compare value to be used in the search.

**character**

Specify the value to compare with the contents of the specified profile parameter. The parameter value must be specified in apostrophes if it contains blanks or special characters.

The *ALWRPL field has two special values: *YES and *NO. When these are specified with the *ALWRPL field, they are changed to internal values for the indicator. When *YES or *NO are specified for the text field, they are used like they are.

The *OWNER field is an 8-character user ID followed by its address. Trailing blanks cannot be omitted from the user ID. For example, if the user ID is JMDOE and the address is SYSTEM1, the query request would be:

```
(*IF (((OWNER *EQ 'JMDOE SYSTEM1'))).
```

If the user ID is JIMSMITH, the query request would be:

```
(*IF (((OWNER *EQ 'JIMSMITHSYSTEM1'))).
```

Dates must be entered in the system date format.

**Note:** If one of the date profile parameters is specified and the compare value is ‘ ’(blank), then the compare value is equated to 01/01/01 or 01/001 for Julian date. ‘/’ is changed depending on the system separator value QDATSEP.

The allowable length for the search fields is limited by the Document Interchange Architecture (DIA) search database. When the length of the value is greater than the maximum, the value is truncated to the allowed length. The maximum lengths are:

**Value**  **Maximum Length**

- *DOCD  44 characters
- *DOCCLS  16 characters
For all operators except *CT and *BG, if a value is specified that is shorter than the profile parameter value, then the specified value is extended with blanks to match the length of the profile parameter.

The case (upper, lower, or mixed) that is used to enter the original parameter value or the case of the comparison value do not matter. The system changes both the entered comparison value and the original parameter value to upper case before making a comparison.

**Element 4: Logical operator**

The logical operators are used to group conditions. The first **AND** operator encountered signifies that a condition group starts with the condition immediately preceding the **AND** operator. Subsequent conditions with the **AND** operator are added to the condition group. The first condition encountered containing the **OR** operator or the last condition in the query definition ends the condition group.

- **AND** The profile parameter value relational groups on both sides of the **AND** value must all be satisfied before a document is selected.

- **OR** If the parameter value relational group on either side of the **OR** value is satisfied, the document is selected.

---

**In folder (FLR)**

Specifies the folders to search for the documents that match the search values specified on the **Query definition (QRYDFN)** parameter and the **Query text (QRYTXT)** parameter.

- **ALL** All the folders on the system are searched.

- **NONE** Documents not located in any folder are searched.
Specify the name of the folders to search for the documents. This is the only folder searched. A folder name can consist of a series of folder names (FLR1/FLR2/etc.) if the documents being searched for are located in a folder contained in another folder. A maximum of 100 folders can be specified and each folder name can be a maximum of 63 characters in length.

**Search subfolder (SCHSUBFLR)**

Specifies whether subfolders of the folder specified on the In folder (FLR) parameter are searched.

*NO  Subfolders are not searched.
*YES Subfolders of the specified folder are searched.

**Query text (QRYTXT)**

Specifies the text search values used to select documents. The values you specify on this parameter are used to search the text index. If values other than *NONE are specified on both the Query definition (QRYDFN) parameter and this parameter, only documents that match both sets of values are selected. If you specify *NONE on both parameters, all documents to which you are authorized are selected.

**Single values**

*NONE  No text search values are entered.

**Element 1: Start query**

*IF  Text search values are used in the document search.

**Element 2: Search criteria**

To specify the conditions under which documents are selected, a set of values is specified for each condition. Each set contains four values:

1. A phrase, which the system compares to entries in the text search index
2. One of the ‘type of matching’ values
3. One of the ‘allow synonyms’ values
4. One of the logical operators

A maximum of 30 sets of values can be specified. Each set must be enclosed in parentheses.

**Element 1: Phrase**

Specifies a phrase which the system compares to entries in the text search index.

**character**

Specify a phrase of one or more words. Do not use any punctuation. When specifying phrases, you can:

* use an asterisk (*) to mask a whole word within a phrase. For example, if you want to search for documents referring to various annual reports, you can specify the phrase:

annual * report
The search results will include documents containing such phrases as annual budget report, annual progress report, and annual sales report. The search results will also include documents containing the phrase ‘annual report’ without a word in between.

When using a word mask, you must specify a word before and after the asterisk. A word mask at the beginning or end of a phrase is ignored.

• use an asterisk (*) to mask part of a word within a phrase. The mask can be used at the beginning, middle, or end of a word. For example, if you want to search for documents referring to word processing, you can specify the phrase:
  
  word process*

  The search results will include documents containing such phrases as word processing, word processor, and word processed.

• use a question mark (?) to mask one or more characters in a word. For example, if you want to search for documents referring to the various spellings of Johnson, you can specify the phrase:
  
  j?hns?n

  The search results will include documents containing such phrases as johnson, johnsen, and jahnson.

Element 2: Type of matching

Specifies a type of matching value to be used in the search.

*ALL  The phrase must be contained within one sentence, but the words do not have to be in the specified order.

*EXACT  The phrase must be contained within one sentence and the words must be in the specified order.

Element 3: Allow synonyms

Specifies whether synonyms are to be used in the search.

*NO  No synonyms are used.

*YES  Synonyms for each word in the phrase, if available, are also used to compare to entries in the text index.

  Note: Using synonyms may affect the performance of the request by causing more words to be searched for, and possibly by causing more documents to be selected.

Element 4: Logical operator

Specifies a logical operator to be used in the search.

*OR  If the phrase on either side of the *OR value is found, the document is selected.

*AND  If the phrases on both sides of the *AND value are found, the document is selected.

*ANDNOT  If the phrase following the *ANDNOT value is not found, the document is selected.
Language ID (TXTLANGID)

Specifies the language identifier for the phrases in the query text. This parameter is required if the Query text (QRYTXT) parameter is specified; it is not allowed if the QRYTXT parameter is not specified or has a value of *NONE.

*JOB The language identifier specified for the job in which this command is entered is used.

c

character Specify a language identifier. Press the PF4 key when prompting the Language ID (TXTLANGID) parameter to see a list of valid identifiers.

Document list (DOCL)

Specifies the name of the document list. The document list contains a pointer to each document in the document library that is qualified for search. This list is a copy of the library at the time the search was run. As documents are deleted from the library or added to the library, the document list is not updated. The document library list name is specified with the name of the user requesting the search.

*DFT A system created name is used as the default name. The default list is the same as the user ID on the User identifier (USRID) parameter.

*NONE No document list is created.

c

character Specify the name of the document list. A maximum of 8 characters can be used.

Text (TEXT)

Specifies the text that briefly describes the object.

*BLANK No text is specified.

c

character Specify a maximum of 50 characters, enclosed in apostrophes.

File to receive output (OUTFILE)

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

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.
Qualifier 2: Library

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

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

name  Specify the name of the library to be searched.

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

Output member options (OUTMBR)

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

Element 1: Member to receive output

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

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

Element 2: Replace or add records

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

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

Type of data for output (OUTDTATYP)

Specifies that certain information about the selected documents is written to the output file if a value is specified on the File to receive output (OUTFILE) parameter.

*DFT  The document information record is written to the output file. This is the same as specifying *DOCD. The record code is 105.

*ALL  All information records about the document are written.

*ACTDATE  The action due date record is written. The record code is 135.

*AUTHOR  The author records are written. The record code is 145.

*CHGDATE  The date last changed record is written. The record code is 130.
*CMPDATE
The completion date record is written. The record code is 140.

*CPYLST
The copy list records are written. The record code is 150.

*CRTDATE
The create date record is written. The record code is 110.

*DOCCLS
The document class record is written. The record code is 155.

*DOCD
The document description record is written. The record code is 105.

*DOCDATE
The document date record is written. The record code is 120.

*EXPDATE
The expiration date record is written. The record code is 115.

*FILCAB
The file cabinet reference record is written. The record code is 160.

*FILDATE
The file date record is written. The record code is 125.

*IDP
The interchange document profile (IDP) is written. The record code is 500.

*IDXDATE
The last indexed date record is written to the output file. OfficeVision/400 text search services must be installed if this value is specified.

*KWD
The keyword records are written. The record code is 170.

*PROJECT
The project record is written. The record code is 185.

*REF
The reference record is written. The record code is 175.

*REVDATE
The date of the last revision to the document content is written to the output file.

*STATUS
The status record is written. The record code is 180.

*SUBJECT
The subject records are written. The record code is 165.

*USEDATE
The date last used record is written. The record code is 200.

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.

Time limit (TIMLMT)

Specifies the amount of time allowed for the requested search to run.

*NOMAX

No time limit for the search is set. All qualified documents are searched.

1-9999

Specify the maximum time limit (in minutes) that the search runs. A two-hour limit is specified as TIMLMT(120). If the search has not been completed when the time limit is reached, the search ends with an informational message followed by a completion message. The output file, and if specified the document list, will contain the documents found within the specified time limit.

Selection limit (SELLMT)

Specifies the allowed number of documents to select in the search.

*NOMAX

No document limit for the search is set. All qualified documents are selected, up to the system limit of 32,767.

1-32,767

Specify the maximum number of documents to select. If there are more documents than the set limit, the document list and the output file contain information about the selected documents up to this limit and an informational message indicating that the limit was reached. The completion message indicates the number of documents selected.

Order by (ORDER)

Specifies that the selected documents are to be ordered (ascending or descending) when placed in the created document list or output file. The order is defined for one or more document profile parameters specified, up to a maximum of 5.

Note: If a value other than *NONE is specified on the Query text (QRYTXT) parameter, ordering is not allowed.

When a user specifies an order to the search request, the performance of the request may be affected. The request performs best if no order is specified.

Single values
No order is applied to the selected documents.

Element 1: Profile

*ACTDATE
The returned documents are ordered by the action due date.

*ASP
The returned documents are ordered by the auxiliary storage pool ID (ASPID) parameter.

*AUTHOR
The returned documents are ordered by the author.

*CHGDATE
The returned documents are ordered by the last changed date.

*CMPDATE
The returned documents are ordered by the completion date.

*CPYLST
The returned documents are ordered by the copy list.

*CRTDATE
The returned documents are ordered by the create date.

*DOCL
The returned documents are ordered by the document description.

*DOCDATE
The returned documents are ordered by the document date.

*DOCTYPE
The returned documents are ordered by the document type profile parameter. Valid values range from 2 through 65535.

*EXDATE
The returned documents are ordered by the expiration date.

*FILDATE
The returned documents are ordered by the date the documents were filed.

*IDXDATE
The returned documents are ordered by the last indexed date profile parameter. Text search services must be installed if this value is specified.

*KWD
The returned documents are ordered by the keyword.

*OWNER
The returned documents are ordered by the name of the owner user profile name.

*PROJECT
The returned documents are ordered by the project.

*REF
The returned documents are ordered by the reference.

*REVDATE
The returned documents are ordered by the last content revision date.

*STATUS
The returned documents are ordered by the status.
The returned documents are ordered by the subject.

The returned documents are ordered by the last used date.

Element 2: Selection order

The returned documents are ordered in the ascending collating sequence.

The returned documents are ordered in the descending collating sequence.

Command character identifier (CMDCHRID)

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

The CMDCHRID parameter applies to the following parameters and means that the data is translated to the code page and character set common to all documents in the search database. That character set and code page is ‘697 500’, except for the User identifier (USRID) parameter, which is ‘930 500’.

The following parameters are translated:

- User identifier (USRID)
- Document list (DOCL)
- Query definition (QRYDFN)
- Query text (QRYTXT)
- Text (TEXT)

Single values

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

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

Specify the graphic character set to use.

Element 2: Code page

Specify the code page to use.
Examples
QRYDOCLIB  USRID(*CURRENT)  OUTFILE(*NONE)  DOCL(MYLIST)
QRYDFN(*IF ((+DOCD  *EQ  DOCDESC  *AND)
(+DOCLCS  *BG  CLASS  *OR)
(+FILDATE  *LE  '06/13/88'))) 

This command searches for documents that meet the following search conditions: document description equals DOCDESC and document class starts with Class; or the file date is on or before 06/13/88. The results of the search will be stored in the document list MYLIST.

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.

CPF905D
 Query of document library failed.

CPF9096
 Cannot use CMDCHRID(*DEVD), DOCCHRID(*DEVD) in batch job.

CPF9860
 Error occurred during output file processing.
Query Distributions (QRYDST)

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

The Query Distribution (QRYDST) command allows you to request distribution information either for yourself or on behalf of another user.

Restrictions:
1. If you request distribution information for another user, you must have been given permission to work on behalf of that user with the GRTUSRPMN command.
2. If user ID(*ALLAUT) is specified, and if you do not have permission to work on behalf of the other user, only information about your own distributions is returned.
3. The requester of the command must be enrolled in the system distribution directory. Personal distribution cannot be requested if the requester is working on behalf of another user.
4. The DLTSTS parameter does not apply to incoming distributions. When OPTION (*IN) is specified, the DLTSTS parameter is ignored.
5. Personal distribution cannot be queried if the requester is working on behalf of another user.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTION</td>
<td>Incoming or outgoing</td>
<td>*IN, *OUT</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>USRID</td>
<td>User identifier</td>
<td>Single values: *CURRENT, *ALLAUT Other values: Element list</td>
<td>Optional, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Element 1: User ID</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Address</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td>DLTSTS</td>
<td>Delete status</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTFILE</td>
<td>File to receive output</td>
<td>Single values: *NONE Other values: Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: File to receive output</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>OUTMBR</td>
<td>Output member options</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Member to receive output</td>
<td>Name, *FIRST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Replace or add records</td>
<td>*REPLACE, *ADD</td>
<td></td>
</tr>
<tr>
<td>CMDCHRID</td>
<td>Command character identifier</td>
<td>Single values: *SYSSVAL, *DEVD Other values: Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Graphic character set</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Code page</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>
Incoming or outgoing (OPTION)

Specifies the type of distribution information that is returned.

*IN  Information about incoming distribution is returned.

*OUT Information about outgoing distribution is returned. An outgoing distribution is the status being saved by the system for a distribution sent to one or more users with confirmation of delivery requested.

User identifier (USRID)

Specifies the user ID and the user ID address associated with this request.

*CURRENT  Your distribution information is returned.

*ALLAUT  Your distribution information and information for all users who have given you permission to work on their behalf is returned.

user-ID  Specify the user ID of the user whose distribution information is being returned. You must have permission to work on behalf of the user specified on this parameter if the named user ID is not your own.

user-ID-address  Specify the user ID address of the user whose distribution information is being returned. You must have permission to work on behalf of the user specified on this parameter if the named user ID address is not your own.

Delete status (DLTSTS)

Specifies whether the distribution status being kept for outgoing distributions is deleted from the system.

*NO  The distribution status is not deleted from the system.

*YES  The distribution status is deleted if all receivers have returned status.

Note: Other products use this status information. Care should be taken not to delete information that other products use to track their distribution.

File to receive output (OUTFILE)

Specifies the name of the database file to which the output is directed. If the output file does not exist, a database file is created in the specified library. If the file is created by this function, the text description will read: ‘OUTFILE is created by QRYDST’, and the authority for users with no specific authority is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library’s create authority.
If a new file is created and *IN is specified on the Incoming or outgoing prompt (OPTION parameter), the system uses QAOSILIN in QSYS with a format name OSLIN as a model.

If a new file is created and *OUT is specified on the Incoming or outgoing prompt (OPTION parameter) the system uses QAOSILOT in QSYS with a format name OSLOUT as a model.

*NONE
The output is not directed to a database file. If *NONE is specified, the output from this command is a completion message containing the number of distributions on the Document Interchange Architecture (DIA) Distribution Recipient Index (*DRX) for the option and user ID and address specified.

When *IN is specified on the Incoming or outgoing prompt (OPTION parameter), the total count of new and old status is returned. After each QRYDST, any mail with new status is changed to old status.

Note: Office Vision/400 disregards the old mail status and treats both old and new mail as if they have new status.

data-base-file-name
Specify the name of the database file and library that receives the output of the display.

The possible library values are:

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

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

library-name
Specify the library where the database file is located.

Output member options (OUTMBR)
Specifies whether the output from the command is displayed at the requesting work station or printed with the job’s spooled output.

The possible member to receive output values are:

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

member-name
Specify the name of the file member that receives the output. If the member does not exist, the system creates the file member. If the member already exists, the system adds records to the end of the member or clears the member and then adds the records.

The possible add or replace values are:

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

*ADD  The system adds the new records to the end of the existing records.
**Status (STATUS)**

Specifies the mail entry status of the distribution for which you are requesting information. This parameter is valid only if *IN is specified on the Incoming or outgoing prompt (OPTION parameter) and an output file is specified on the File to receive output prompt (OUTFILE parameter).

- **ALL** Distribution information is returned regardless of the distributions’ mail entry status.
- **NEW** Distribution information is returned only for those distributions whose mail entry status is NEW.
- **OLD** Distribution information is returned only for those distributions whose mail entry status is OLD, which indicates that the distribution has been queried once but has not been processed.
- **OPENED** Distribution information is returned only for those distributions whose mail entry status is OPENED.
- **UNOPENED** Distribution information is returned for those distributions whose mail entry status is either NEW or OLD.
- **LOCAL** Distribution information is returned only for those distributions whose mail entry status is LOCAL, which indicates that the distribution has been filed on the local system.
- **REMOTE** Distribution information is returned only for those distributions whose mail entry status is REMOTE, which indicates that the distribution has been filed on a remote system.
- **FILEPND** Distribution information is returned only for those distributions whose status is FILEPND, which indicates that the distribution is being filed on the local system or a remote system, but the file operation has not finished.
- **DELETED** Distribution information is returned only for those distributions whose status is DELETED, which indicates that the document referred to by the distribution has been deleted.
- **DAMAGED** Distribution information is returned only for those distributions whose status is DAMAGED, which indicates that the document referred to by the distribution is damaged.

**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 prompt (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.

*DEVD
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.

Examples
QRYDST USER(*CURRENT) OPTION(*IN)
OUTFILE(*CURLIB/MYFILE) OUTMBR(*FIRST *ADD)

This command requests information about incoming distributions for the current user of this command. The output is directed to the database file MYFILE in the user’s current library and is added to the first member in that file.

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.

CPF907C
&1 requested distributions completed, acknowledge failed.

CPF9096
Cannot use CMDCHRID(*DEVD), DOCCHRID(*DEVD) in batch job.

CPF9097
Query distribution request failed.

CPF9845
Error occurred while opening file &1.

CPF9847
Error occurred while closing file &1 in library &2.
CPF9860
Error occurred during output file processing.
Query Problem Status (QRYPRBSTS)

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

The Query Problem Status (QRYPRBSTS) command retrieves problem status information from ‘IBMSRV (RETAIN) or from another AS/400 system that is enlisted as a service provider.

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

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRBID</td>
<td>Problem identifier</td>
<td>Character value, *PMR</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>ORIGIN</td>
<td>Origin</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Network</td>
<td>Communications name, *NETATR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>identifier</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Control point name</td>
<td>Communications name, *NETATR</td>
<td></td>
</tr>
<tr>
<td>RMTCPNAME</td>
<td>Remote control point</td>
<td>Communications name, *IBMSRV, *SELECT</td>
<td>Optional</td>
</tr>
<tr>
<td>RMNENETID</td>
<td>Remote network identifier</td>
<td>Communications name, *NETATR</td>
<td>Optional</td>
</tr>
<tr>
<td>SRVID</td>
<td>Service number</td>
<td>Character value</td>
<td>Optional</td>
</tr>
<tr>
<td>AUTOPRBCRT</td>
<td>Auto problem create</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Problem identifier (PRBID)

Specifies the problem identifier of the problem log entry. Problems with different system origins can have the same identifier. This parameter can be used with the ORIGIN parameter to select a single problem from a particular system origin.

*PMR  *PMR special value is used when the service request is sent to IBM service support. SRVID must be specified when PRBID is *PMR.

Specify a problem identifier.

Specify a problem identifier.

This is a required parameter.
Origin (ORIGIN)

Specifies the node of the system from which the problem log entry originated. This parameter is used with the PRBID parameter to uniquely identify the problem.

The possible Network Identifier values are:

*NETATR

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

network-identifier

Specify a network identifier.

The possible Control Point Name values are:

*NETATR

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

control-point-name

Specify a control point name.

Remote control point (RMTCPNAME)

Specifies the destination of the service provider to whom the service request is sent.

The possible Remote Control Point Name values are:

*IBMSRV

The service request is sent to IBM service support.

*SELECT

A list of service providers is shown from which the user can select the destination the service request is sent to.

remote-control-point-name

Specify the name of the control point that is the destination of the request.

Remote network identifier (RMTNETID)

Specifies the remote name of the service provider’s network.

*NETATR

The service provider is in the local network.

remote-network-identifier

Specify the network name of the service provider to whom the request is sent.

Service number (SRVID)

Specifies the service assigned number for the problem log entry. This number was assigned when the problem was reported to IBM service support.
**service-assigned-number**
Specify the service assigned number for the problem log entry.

---

**Auto problem create (AUTOPRBCRT)**

Specifies whether a problem should automatically be created, if a problem does not exist on the system.

*YES  Create a problem.
*NO   Do not create a Problem.

---

**Examples**

**Example 1: Querying Problem Status on Another System**

```
QRYPRBSTS  PRBID(1234567890)  RMTCPNAME(SYSTEM99)
            RMTNETID(IBMNETID)  AUTOPRBCRT(*YES)
```

This command searches for the status of a specific problem on another system (SYSTEM99).

**Example 2: Querying IBM Service**

```
QRYPRBSTS  PRBID(*PMR)  RMTCPNAME(IBMSRV)  RMTNETID(*NETATR)
            AUTOPRBCRT(*YES)
```

This command searches the IBM Service database for the status of PMR 8X123.

---

**Error messages**

**ESCAPE Messages**

CPF7AA7
Problem &1 not found or in use.

CPF7AD4
Network ID &1 not in correct format.

CPF7A84
Query status request routed to different system than specified.

CPF7A88
Error indicated in reply to request.

CPF7A9A
Remote control point and network identifier not valid.

CPF7A9B
Problem &1 cannot be queried.

CPF7A97
Invalid service identifier.

CPF7A98
Service identifier not allowed.
CPF7A99
Query must be sent to *IBMSRV.

CPF7B18
Control point &1 not in correct format.

CPF8C08
Cannot specify *SELECT for the control point name.

CPF8C09
&1 not defined as a service provider.

CPF8C24
Error occurred while processing request.

*STATUS Messages
CPZ7A80
Sending query status request to &1.
Query TIE Files (QRYTIEF)

Where allowed to run:
- Batch job ("BATCH")
- Batch program ("BPGM")
- Batch REXX procedure ("BREXX")
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API ("EXEC")

Threadsafe: No

The Query Technical Information Exchange File (QRYTIEF) command allows you to find out whether files are ready to be received from the remote support network. A message is returned that specifies the size of the largest file that is to be received.

There are no parameters for this command.

Parameters
None

Examples
QRYTIEF

This command sends a message that specifies the number of files to be received from the remote support network and the size of the largest file to be received.

Error messages
None
Start QSH (QSH)

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

The Start QSH (STRQSH) command, as known as QSH, starts the qsh shell interpreter.

If run in an interactive job, STRQSH starts an interactive shell session. If a shell session is not already active in the job, then:
1. A new shell session is started and a terminal window is displayed.
2. qsh runs the commands from the file /etc/profile if it exists.
3. qsh runs the commands from the file .profile in the user’s home directory if it exists.
4. qsh runs the commands from the file specified by the expansion of the ENV variable if it exists.

If a shell session is already active in an interactive job, you are reconnected to the existing session. From the terminal window, you can enter shell commands and view output from the commands.

Using the Terminal Window

The terminal window has two parts:
• an input line for entering commands, and
• an output area that contains an echo of the commands you entered and any output generated by the commands.

The terminal window supports the following function keys:
F3 (Exit)
Close the terminal window and end the qsh session.

F5 (Refresh)
Refresh the output area.

F6 (Print)
Print the output area to a spool file.

F7 (Up)
Roll output area up one page.

F8 (Down)
Roll output area down one page.

F9 (Retrieve)
Retrieve a previous command. You can press this key multiple times to retrieve any previous command. For example, to retrieve the second to last command you entered, press this key two times. You can also select a specific command to be run again by placing the cursor on that command and pressing this key. When the interactive job is running in a double-byte CCSID, this key is not available.

F11 (Toggle line wrap)
Toggle the line wrap/truncate mode in the output area. In line wrap mode, lines longer than the width of the terminal window are wrapped to the next line. In truncate mode, the portion of a line beyond the width of the terminal window is not shown.
F12 (Disconnect)  
Disconnect from the qsh session. This key only closes the terminal window and does not end the qsh session. You can redisplay the disconnected qsh session by running STRQSH again.

F13 (Clear)  
Clear the output area.

F17 (Top)  
Display top of output area.

F18 (Bottom)  
Display bottom of output area.

F19 (Left)  
Shift output area to the left.

F20 (Right)  
Shift output area to the right.

F21 (CL command entry)  
Display a command entry window where you can enter CL commands.

Also, you can use SysReq 2 to interrupt the currently running command.

Error messages for STRQSH

*ESCAPE Messages

QSH0002  
Error found with QSH session, reason code &1, errno &2.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMD</td>
<td>Command</td>
<td>Character value, *NONE</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Command (CMD)

Specifies the shell command to be run.

The possible values are:

*NONE:  
No command is provided and an interactive session is started.

command  
A shell command to run. The command can be a maximum of 5000 bytes in length. If a blank or other special characters are used, the command must be enclosed in apostrophes ('). If an apostrophe is intended, two apostrophes must be used ("').

Note: The case is preserved when lowercase characters are specified.
Examples
None

Error messages
*ESCAPE Messages
QSH0002
   Error found with QSH session, reason code &1, errno &2.
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPHT (Merge TCP/IP Host Table)
Qualifier Definition (QUAL)

The Qualifier (QUAL) command definition statement describes one part of a qualified name. If a name is the allowable value of a parameter or list item defined in a PARM or ELEM statement, it can be changed to a qualified name by using a QUAL statement for each qualifier used to qualify the name.

The order in which the QUAL statements are entered into the source file determines the positional order in which the qualifiers must be specified and passed to the validity checker and the command processing program. The first qualification of a qualified name must be either a simple name, a generic name, or a defined special value.

The QUAL statement (or only the first QUAL statement if there are more than one) must have a statement label that matches the statement label value that must be specified in a PARM or ELEM statement for which the qualifier is being defined. The qualifiers for the parameter or list item are then entered on the command in the form: value3/value2/value1, where values 1 through 3 are qualifiers that are each described by a QUAL statement. The values are passed to the command processing program in the same order, with the periods removed, and with each value padded to its maximum length.

Note: The QUAL statement contains certain parameters and predefined values that can be used only when IBM-supplied command processing programs are called by the command being defined. Because of limitations in some high-level languages, these values may not be useful in the definition statements of user-defined commands. These parameters and values are identified by the phrase (For IBM-supplied commands) that immediately follows the parameter keyword (if the entire parameter is for IBM-supplied commands only) or the predefined value to which it applies.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEN</td>
<td>Length specification</td>
<td>Element list</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>Element 1: Value length</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSTANT</td>
<td>Constant value</td>
<td>Character value</td>
<td>Optional</td>
</tr>
<tr>
<td>RSTD</td>
<td>Restricted values</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>DFT</td>
<td>Default value</td>
<td>Character value</td>
<td>Optional</td>
</tr>
<tr>
<td>VALUES</td>
<td>Valid values</td>
<td>Values (up to 300 repetitions): Character value</td>
<td>Optional</td>
</tr>
<tr>
<td>REL</td>
<td>Relational expression</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td>Element 2: Value</td>
<td>Character value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RANGE</td>
<td>Range of values</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td>Element 1: Lower value</td>
<td>Character value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element 2: Upper value</td>
<td>Character value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
<td>----------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>SPCVAL</td>
<td>Special values</td>
<td>Values (up to 300 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: From value</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: To replacement value</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td>Minimum values required</td>
<td>0, 1</td>
<td>Optional</td>
</tr>
<tr>
<td>ALWUNPRT</td>
<td>Allow unprintable characters</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>ALWVAR</td>
<td>Allow variable names</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>FULL</td>
<td>Full field required</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>EXPR</td>
<td>Value an expression</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>VARY</td>
<td>Varying length</td>
<td>Single values: *NO</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Return length value</td>
<td>*YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Value length</td>
<td>*INT2, *INT4</td>
<td></td>
</tr>
<tr>
<td>PASSATR</td>
<td>Pass attribute byte</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>DSPINPUT</td>
<td>Display input</td>
<td>*YES, *PROMPT, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>CHOICE</td>
<td>Choice text</td>
<td>Character value, *VALUES, *NONE, *PGM</td>
<td>Optional</td>
</tr>
<tr>
<td>CHOICEPGM</td>
<td>Choice program</td>
<td>Single values: *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Choice program</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>INLPMTLEN</td>
<td>Initial prompt length</td>
<td>*CALC, *PWD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 17, 25, 32, 50, 80, 132, 256, 512</td>
<td>Optional</td>
</tr>
<tr>
<td>PROMPT</td>
<td>Prompt text or message ID</td>
<td>Character value, *NONE</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**Type of value (TYPE)**

Specifies the type of qualifier used to qualify a parameter name or list element name. The qualifier can be a name or generic name, a quoted or not quoted character string, or an integer. Enter one of the following options to specify the type of qualifier. The first qualifier for any qualified name must have a type of name (*NAME) or generic name (*GENERIC).

**NAME**

The qualifier is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic or one of the special characters $, @, or #. The remaining characters can be alphanumeric, a period, an underscore, or one of special characters $, @, or #. The name can also be a string of characters starting and ending with double quotation marks (") or enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it should be specified on the Special values (SPCVAL) parameter.

**SNAME**

The qualifier is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic or one of the special characters $, @, or #. The remaining characters can be alphanumeric, an underscore, or one of the special characters $, @, or #. The character string can be enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it must be specified on the Special values (SPCVAL) parameter.

**CNAME**

The qualifier is a character string that represents a name. The maximum length of the name is 256 characters. The first character must be alphabetic or one of the special characters $, @, or #. The...
remaining characters can be alphanumeric or one of special characters, $, @, or #. The character string can be enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it must be specified on the Special values (SPCVAL) parameter.

*GENERIC
The qualifier is a character string that represents a generic name. A generic name contains a maximum of 255 characters followed by an asterisk (*); the name identifies a group of objects whose names all begin with the characters preceding the asterisk (*). If an asterisk is not included, the system assumes that the generic name is a complete object name.

*CHAR
The qualifier is a character string that can (optionally) be enclosed in apostrophes. If the character string contains any special characters (not including an asterisk (*)), it must be enclosed in apostrophes. The maximum length of the character string is 5000 characters.

*INT2
The qualifier is an integer that is passed as a 2-byte signed binary number.

*INT4
The qualifier is an integer that is passed as a 4-byte signed binary number.

*UINT2
The qualifier is an integer that is passed as a 2-byte unsigned binary number.

*UINT4
The qualifier is an integer that is passed as a 4-byte unsigned binary number.

---

**Length specification (LEN)**

Specifies the length of the qualifier, if *NAME, *GENERIC, or *CHAR is specified for the Type of value (TYPE) parameter.

---

**Constant value (CONSTANT)**

Specifies that a value is passed to the command processing program as a constant for the qualifier when the command being defined is processed; the qualifier is not to appear externally on the command. If specified, the value must satisfy the requirements specified by the following parameters:

- Type of value (TYPE parameter)
- Length specification (LEN parameter)
- Valid values (VALUES parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)

If a character constant is specified in this parameter, it can be no longer than 32 characters.

If a constant is specified in this QUAL statement and other QUAL statements immediately follow it, they must also be defined as constants, unless a label precedes one of them. A label indicates the beginning of a new group of QUAL statements, which can be defined differently.

Also, if a constant is specified for the qualifier being defined, no prompt text can be specified for the Prompt text or message ID (PROMPT) parameter of this QUAL statement. However, any other qualifiers or groups of qualifiers are still prompted, and their values are still passed to the command processing program as a qualified name.
This parameter is not valid if the Default value (DFT) parameter is specified or if *YES is specified for the Value an expression (EXPR) parameter.

Variables cannot be coded for this parameter.

---

**Restricted values (RSTD)**

Specifies whether the value entered for the qualifier is restricted to only one of the values given in the Valid values (VALUES) parameter or the Single values (SNGVAL) parameter, or whether any value can be used that satisfies the requirements specified by the following parameters:

- Type of value (TYPE parameter)
- Length specification (LEN parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)

**NO** The value entered for the qualifier defined by this QUAL statement can be anything that satisfies the requirements specified by the following parameters:

- Type of value (TYPE parameter)
- Length specification (LEN parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)

**YES** The value entered for the qualifier in this QUAL statement is restricted to one of the values in the Valid values (VALUES) parameter, or to one of the from-values in the Special values (SPCVAL) parameter.

---

**Default value (DFT)**

Specifies the default value assigned to the qualifier if a value is not specified by the user. The default value must satisfy one of the following:

- It must match the qualifier requirements specified by the following parameters:
  - Type of value (TYPE parameter)
  - Length specification (LEN parameter)
  - Relational expression (REL parameter)
  - Range of values (RANGE parameter)

- It must be one of the from-values in the Special values (SPCVAL) parameter.
- If *YES is specified for the Restricted values (RSTD) parameter, it must be in the list of values in the Valid values (VALUES) parameter or in the list of from-values in the Special values (SPCVAL) parameter.
- If the default is a character constant, it can have no more than 32 characters.

This parameter is valid only if the Minimum values required (MIN) parameter is 0, which means the qualifier defined by this QUAL statement for this list is optional. A default is not meaningful on this QUAL statement if it is the first one (defining the first part) for a qualified name and if a default is specified on the PARM or ELEM statement that this QUAL statement further defines.
not specified, it has a default of its own: the default is blank if *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC is specified for the Type of value (TYPE) parameter. The default is zero (0) if *INT2, *INT4, *UINT2 or *UINT4 is specified for the Type of value (TYPE) parameter. An assumed default value is not displayed by the command prompt; a blank input field is shown instead. If a default is specified in this parameter, it is displayed by the prompt exactly as specified.

The DFT parameter is not valid if the Constant value (CONSTANT) parameter is specified.

value  Specify the default value that meets the specified requirements or that is one of the values specified in the Valid values (VALUES) parameter or the Special values (SPCVAL) parameter.

Variables cannot be coded for this value.

Valid values (VALUES)

 Specifies a list of up to 300 constants (fixed values) from which one constant can be entered as the value of the qualifier. This parameter is valid only if all of the following are true:

• *YES is specified for the Restricted values (RSTD) parameter.
• Both the Range of values (RANGE) parameter, and the Relational expression (REL) parameter are not specified,
• The constant matches the attributes specified by the Type of value (TYPE) parameter, and the Length specification (LEN) parameter in this QUAL statement.

Character constants specified in this parameter can be no longer than 32 characters. Specify the constants (not more than 300) that can be entered as the value of the qualifier.

Relational expression (REL)

 Specifies the relationship between the qualifier value and the value of another parameter or constant. To specify the relationship, enter one of the following relational operators followed by a constant or the value of another parameter.

*LT less than
*LE less than or equal to
*EQ equal to
*GE greater than or equal to
*GT greater than
*NL not less than
*NE not equal to
*NG not greater than

This parameter is not valid if either the Valid values (VALUES) parameter or the Range of values (RANGE) parameter is specified. If *CHAR (character type) is specified by Type of value (TYPE) parameter, the EBCDIC value of the character string is used as an unsigned integer in the comparison. If a character constant is specified in this parameter, it can be no longer than 32 characters.

Qualifier Definition (QUAL)  379
Range of values (RANGE)

Specifies the range, or limits, for the value of the qualifier. The qualifier value must be greater than or equal to the lower limit value specified, and it must be less than or equal to the upper limit value specified. For nonnumeric data types, such as *CHAR, the range of values and the data specified is right-justified and padded on the left with blanks. A numeric range should not be used to define an interval for nonnumeric data unless leading zeros are specified or the data is only 1 character in length. This parameter is not valid if either the Valid values (VALUES) parameter, or the Relational expression (REL) parameter is specified. Character constants specified in this parameter can be no longer than 32 characters.

Special values (SPCVAL)

Specifies a list of up to 300 entries that define special values that can be entered on the parameter named in the Keyword (KWD) parameter on the PARM statement. Each entry specifies a character string (a from-value) that can be entered even though it may not meet all validity checking requirements. If the entered character string matches the from-value of one of the entries, and the to-value is specified, the string is replaced with the to-value and is then passed to the command processing program without further checking. If the to-value is omitted, the from-value is passed to the command processing program. The from-value is a character string, but the to-value can be anything that is passable. If a CL variable is used for the from-value, its type must be *CHAR. However, the first qualifier can only have special to-values with the from-values that are a name, a generic name, or an asterisk (*) followed by a name such as *ALL.

Each to-value must be passable to the command processing program. The to-value must be no longer than is specified on the Length specification (LEN) parameter and, if *INT2, *INT4, *UINT2 or *UINT4 is specified for the Type of value (TYPE) parameter, the type of the to-value must be the same. If a character type (such as *CHAR or *NAME) is specified for the Type of value (TYPE) parameter, the to-value must be a character string. Character constants specified in this parameter can be no longer than 32 characters. If a to-value is not specified, the from-value must be passable.

If a to-value of *CURLIB is specified, the name of the current library is passed to the command processing program rather than the value *CURLIB. If the from-value is *CURLIB and no to-value is specified, or if the to-value is *CURLIB and it is enclosed in apostrophes, the value *CURLIB is passed to the command processing program.

Variables cannot be coded for this value.

Minimum values required (MIN)

Specifies whether the qualifier being defined in this QUAL statement is required or optional. If this parameter is not specified, 0 is assumed, which means the qualifier is optional. If a required qualified name is needed, 1 must be specified for this parameter on both the first QUAL and on the PARM or ELEM that refers to it.

0 The qualifier is optional on the name being qualified.

1 The qualifier is required on the name being qualified; it must be entered.
Allow unprintable characters (ALWUNPRT)

Specifies whether this QUAL statement should accept the hexadecimal characters above X’FF’ or those in the range of X’00’ to X’3F’. This parameter is valid only if *CHAR or *X is specified for the Type of value (TYPE) parameter.

*YES All characters can be sent to the display or printer.
*NO Unprintable characters cannot be passed to the command processing program.

Allow variable names (ALWVAR)

Specifies whether variable names are allowed for the qualifier. *NO is not allowed if *VARNAME, *ZEROELEM, *NULL, or a statement label is specified for the Type of value (TYPE) parameter.

*YES Variable names can be used for the qualifier.
*NO Variable names cannot be used for the qualifier.

Full field required (FULL)

Specifies whether the number of characters in the qualifier value must be exactly the same as the number specified in the Length specification (LEN) parameter (if specified) or its default length (if LEN is not specified).

*NO The number of characters in the qualifier value can be less than that specified by the Length specification (LEN) parameter.
*YES The number of characters in the qualifier value must equal the number specified by the Length specification (LEN) parameter or the default length for that type. The exact length is valid only if *CHAR, *NAME, or *GENERIC is specified for the Type of value (TYPE) parameter.

Value an expression (EXPR)

Specifies whether the qualifier can accept an expression containing a character concatenation.

*NO The qualifier value cannot be a concatenation expression.
*YES The qualifier value can be a concatenation expression.

Varying length (VARY)

Specifies whether the qualifier value that is passed to the command processing program is preceded by a length value that indicates the number of characters entered for the qualifier’s value.

Single values

*NO The qualifier value is not preceded by a length value.

Element 1: Return length value
*YES The qualifier value passed to the command processing program is preceded by a binary length field that indicates the number of characters actually specified for the qualifier. *YES is valid only if *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC is specified for the Type of value (TYPE) parameter. *YES must be specified if PASSATR(*YES) and RTNVAL(*YES) are specified.

Note: The length value is the actual number of characters entered for the command parameter with trailing blanks removed. The length value passed may be different than the defined parameter length or the declared variable length. The length of the field containing the character string data is determined by the defined length for the parameter or the declared LEN for CL Program variables. The length value defines how many characters in the character string data field were actually entered for the command parameter.

Element 2: Value length

*INT2 The qualifier value is an integer passed as a 2-byte signed binary number.

*INT4 The qualifier value is an integer passed as a 4-byte signed binary number.

---

Pass attribute byte (PASSATR)

Specifies whether an attribute byte is to be passed to the command processing program with the qualifier. The attribute byte precedes the qualifier data.

*NO No attribute byte is passed with the qualifier.

*YES An attribute byte is passed with the qualifier.

The attribute byte has two fields:

1. The leftmost bit of the attribute byte indicates whether or not a value was specified. If the leftmost bit is '0'B, the value passed to the command processing program is a default value and was not specified in the command string. If the leftmost bit is '1'B, the value passed to the command processing program was specified in the command string.

2. The remaining seven bits describe the value passed to the command processing program when *CHAR is specified for the Type of value (TYPE) parameter.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'0000010'B</td>
<td>Meets *NAME rules, like A_B</td>
</tr>
<tr>
<td>'0000010'B</td>
<td>Meets <em>GENERIC rules, like AB</em></td>
</tr>
<tr>
<td>'1000101'B</td>
<td>Quoted character string, like 'A B'</td>
</tr>
<tr>
<td>'0000101'B</td>
<td>Unquoted character string, like 5A</td>
</tr>
<tr>
<td>'1001000'B</td>
<td>Logical constant, '0' or '1'</td>
</tr>
<tr>
<td>'0001100'B</td>
<td>Hexadecimal value, like X'C1C2'</td>
</tr>
<tr>
<td>'0100001'B</td>
<td>Unsigned numeric value, like 5</td>
</tr>
<tr>
<td>'0101001'B</td>
<td>Unsigned numeric with decimal point, like 5.2</td>
</tr>
<tr>
<td>'0110001'B</td>
<td>Signed numeric value, like -5</td>
</tr>
<tr>
<td>'0111001'B</td>
<td>Signed numeric with decimal point, like -5.2</td>
</tr>
</tbody>
</table>

---

Display input (DSPINPUT)

Identifies whether the keyword value is to be shown in the job log or in a prompt display.

*YES Indicates that the parameter value is shown on the prompt display and in the job log.
*PROMPT
The response *PROMPT indicates that the parameter value is shown on the prompt display but not in the job log.

*NO Indicates that the parameter value is not shown on either the prompt display or in the job log.

Choice text (CHOICE)
Specifies the choices text that is displayed to the right of the input field on the prompt screen. Up to 30 characters of text can be displayed.

*VALUES
The choices text is generated based on the values specified for the TYPE, RSTD, RANGE, SNGVAL, SPCVAL, and VALUES parameters. If constants are specified for the RANGE parameter, the choices text begins with the minimum value and the maximum value separated by a hyphen. If RANGE is not specified with constants as the minimum and maximum values, and RSTD(*NO) is specified, the choices text begins with a short description of the parameter type based on the value specified for the TYPE parameter. Values specified for the SNGVAL parameter are added to the choices text, in the order the values are defined in the command definition source and separated by a comma and a blank. The last entries added to the choices text are values specified for the SPCVAL or VALUES parameter, in the order the values are defined in the command definition source and separated by a comma and a blank. If there are too many values to fit in 30 characters, the last value is followed by three periods.

The following are examples of possible choices text generated by CHOICE(*VALUES):
- If TYPE(*DEC) and RANGE(1.0 999.9) and SPCVAL(*NOMAX -1) are specified, the choices text will be:
  1.0-999.9, *NOMAX
- If TYPE(*NAME) and RSTD(*NO) and SNGVAL(*ALL) and SPCVAL(*LIBL *CURLIB) are specified, the choices text will be:
  Name, *ALL, *LIBL, *CURLIB
- If RSTD(*YES) and SNGVAL(*ALL) and SPCVAL(*ALRTBL *BNDDIR *CHTFMT *CLD *CLS *CMD) are specified, the choices text will be:
  *ALL, *ALRTBL, *BNDDIR...

*NONE
No values are displayed.

*PGM A program that is called determines the values that are displayed. The program that is called is identified in Choice program (CHOICEPGM) parameter of the PARM statement.

message-identifier
Specify the message ID of the message used to retrieve the message containing the text for the possible values field. The message file specified on the Message file for prompt text (PMTFILE) parameter of the Create Command (CRTCMD) command is used to find the message.

'choices-text'
Specify no more than 30 characters, enclosed in apostrophes.
Choice program (CHOICEPGM)

Specifies the program to be called during command prompting to fill in the possible choices text and the permissible values. This parameter must be specified if *PGM is specified on the Choice text (CHOICE) parameter and may not be specified otherwise.

Single values

*NONE  No program is identified to fill in the possible choices text and permissible values.

Qualifier 1: Choice program

name  Specifies the name of the program to be called during prompting to fill in the possible choices text or permissible values. If an exception occurs when the program is called, no possible choices text is left blank, and the list of permissible values is taken from the command.

Qualifier 2: Library

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

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

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

Initial prompt length (INLPMTLEN)

Specifies the length of the input field initially displayed for the qualifier when the command is prompted. The user can extend the field to a maximum length of 512 bytes by entering an ampersand (&) in the first position of the field, followed by a blank. INLPMTLEN is valid only if TYPE is specified as *NAME, *SNAME, *CNAME, *GENERIC, or *CHAR. If FULL(*YES), RSTD(*YES), or CONSTANT are specified, INLPMTLEN(*CALC) must be specified or defaulted.

*CALC  The prompter will determine the length of the prompt field based on the type and length of the parameter.

*PWD  If the current value of system value QPWDLVL is ‘O’ or ‘1’, the prompt field will be 10 bytes long. Otherwise, the length of the prompt field will be determined by the length of the parameter. INLPMTLEN(*PWD) is valid only if TYPE is specified as *CHAR, *NAME, *SNAME, or *CNAME.

initial-prompt-length

Specify the initial length in bytes. Valid values are 1-12, 17, 25, 32, 50, 80, 132, 256, and 512.

Prompt text or message ID (PROMPT)

Specifies the prompt text, if any, that is used for the qualifier (defined in this QUAL statement). This parameter is not allowed for the first qualifier or for a qualifier for which the Constant value (CONSTANT) parameter is specified. The prompt text for the first qualifier comes from the PROMPT parameter of the PARM or ELEM statement pointing to the qualifier. The prompt text gives a short description of the qualifier which appears next to the qualifier input field when the command is prompted.
*NONE

No prompt text is shown for the qualifier defined by this QUAL statement. This qualifier is still prompted by an input field, but no text is shown with it.

**message-identifier**

Specify the message identifier that specifies the message containing the prompt text of up to 30 characters that is shown when the program is prompting the qualifier. If a message having the specified identifier cannot be found in the message file specified in the **message file for prompt text** (PMTFILE) parameter of the Create Command (CRTCMD) command, the message identifier itself is used as the prompt text.

*prompt-text*

Specify the prompt text that is shown when the program is prompting the qualifier. The text must be a character string of no more than 30 characters, enclosed in apostrophes.

---

**Examples**

**Example 1: Qualified Job Name as One Element**

```plaintext
PARM       KWD(SPLFILE) TYPE(L1) DFT(*) SNGVAL(*)
L1:        ELEM TYPE(*NAME) MIN(1) /* For file name */
           ELEM TYPE(Q1)
Q1:        QUAL TYPE(*NAME) MIN(1) /* For job name */
           QUAL TYPE(*NAME) /* For user name */
           QUAL TYPE(*CHAR) LEN(6) /* For job number */
```

The SPLFILE parameter is optional and, if not specified, defaults to an asterisk (*). Otherwise, the value consists of a two-element list. The first element is a file name and it is required. The second element is a qualified job name. The first qualifier is required; the last two qualifiers are optional. The following are some examples of valid SPLFILE parameter syntax:

- SPLFILE(*)
- SPLFILE(MYSPLFILE MYJOB)
- SPLFILE(MYSPLFILE 123456/USERA/MYJOB)

**Example 2: List of Qualified Object Names as One Element**

```plaintext
PARM       KWD(DTAMBRS) TYPE(L1) DFT(ALL) MAX(32) + SNGVAL(*ALL)
L1:        ELEM TYPE(Q1) MIN(1)
           ELEM TYPE(*NAME) MIN(0) MAX(32) SPCVAL(*NONE) + DFT(*NONE)
Q1:        QUAL TYPE(*NAME) MIN(1)
           QUAL TYPE(*NAME) DFT(*CURRENT) SPCVAL(*CURRENT)
```

The parameter named DTAMBRS is optional and, if not specified, defaults to *ALL. Otherwise, the value consists of a list, each element of which is itself a list. Each sublist consists of a qualified file name optionally followed by one or more member names. If no member name is specified, *NONE is taken as the default. If no library qualifier is specified for the file, *CURRENT is taken as the default. Each sublist can contain one file name and up to 32 member names. Up to 32 such sublists can appear as the value of DTAMBRS. The following are some examples of valid DTAMBRS parameter syntax:

- DTAMBRS(*ALL)
- DTAMBRS((PFILE1 *NONE))
- DTAMBRS((LIB1/PFILE1 (MBR1 MBR2)))
- DTAMBRS((*CURRENT/PFILE1 (MBR1 MBR2 MBR3)) (LIB2/PFILE2 (MBRA MBRB)))
Error messages

None
Reclaim Activation Group (RCLACTGRP)

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

The Reclaim Activation Group (RCLACTGRP) command deletes a specified activation group and frees the resources that are scoped to it. It does not reclaim resources scoped to the job or scoped to the default activation group. This command is normally used only in the controlling program of the application.

An activation group is eligible to be reclaimed if it meets the following criteria:
- The activation group is not the default activation group.
  The default activation group cannot be reclaimed.
- The activation group is not active.
  An activation group cannot be reclaimed if there are programs or procedures running within the activation group.
- The activation group is not one of the debug activation groups.
  When the job is in debug mode, the activation groups in use do not appear as active on the Call Stack or Display Activation Group displays.
- The activation group is not a shared activation group.
  A shared activation group cannot be reclaimed because it may be in use by another job.

When an activation group is reclaimed, all resources within the scope of the activation group are reclaimed. Resources within the scope of the activation group include static storage for programs in the activation group, open files, user interface manager (UIM) application resources, Common Programming Interface (CPI) Communications conversations, hierarchical file systems (HFS) resources, user-defined communications sessions, and pending changes for the commitment definition.

A close option can be specified on this command, and is used when closing mixed, communications, binary synchronous (BSC), and intersystem communications function (ICF) files. If an activation group level commitment definition has been started for the activation group, and it has pending committable changes, the close option also indicates whether the system implicitly commits or rolls back the pending changes before ending the commitment definition. When specifying a close option of *NORMAL, and there are no errors when closing files using the activation group level commitment definition, a commit is performed. Otherwise, a rollback is performed. See the Backup and Recovery book, SC41-5304 book for information on how the system performs the rollback operation under commitment control.

An activation group should only be reclaimed if it will never be needed again within the same job. Otherwise, errors and unpredictable results may occur if other programs later attempt to access the resources that were reclaimed. Therefore, this command should normally only be used in the controlling program of an application. Specifying ACTGRP(*ELIGIBLE) requires full knowledge of the job environment. Otherwise, unpredictable results can occur.

For more information on the appropriate use of this command, see the IBM System Manager for i5/OS.

Restrictions:
1. This command is not threadsafe. This command fails in a job that allows multiple threads.
Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTGRP</td>
<td>Activation group</td>
<td>Name,*ELIGIBLE</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>OPTION</td>
<td>Close option</td>
<td>*NORMAL, *ABNORMAL</td>
<td>Optional, Positional 2</td>
</tr>
</tbody>
</table>

Activation group (ACTGRP)

Specifies the activation group to be reclaimed.

*ELIGIBLE

All eligible activation groups within the job are deleted.

.activation-group-name

Specify the activation group to be reclaimed. The activation group can only be reclaimed if it has no active calls. If active calls exist, a message is displayed informing the user that the request failed. If the activation group is not found, a message is displayed informing the user that the request failed because the activation group was not found.

Close option (OPTION)

Specifies whether to commit or roll back pending changes for an activation group level commitment definition, and whether a normal or abnormal close notification is sent to the attached host system when mixed, communications, BSC, and ICF files are closed. This parameter is ignored for all other files and objects within the scope of the activation group.

*NORMAL

The changes pending for an activation group level commitment definition are committed (if there are no errors when closing files using the commitment definition), and a normal close notification is sent to the attached host system when mixed, communications, BSC, and ICF files are closed.

*ABNORMAL

The changes pending for an activation group level commitment definition are rolled back and an abnormal close notification is sent to the attached host system when mixed, communications, BSC, and ICF files are closed.

Examples

RCLACTGRP ACTGRP(MYGROUP)

This command reclaims the activation group MYGROUP.

Error messages

*ESCAPE Messages
CPF1653
   Activation group &1 not found.

CPF1654
   Activation group &1 cannot be deleted.

CPF180C
   Function &1 not allowed.

CPF1892
   Function &1 not allowed.
Reclaim DB Cross-Reference (RCLDBXREF)

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

The Reclaim Database Cross-Reference (RCLDBXREF) command recovers the database cross-reference catalog data for a specific library. This command provides a subset of the function provided by the Reclaim Storage (RCLSTG) command to reclaim the database cross-reference catalog data for the system by specifying SELECT(*DBXREF). The difference is that this command does not require the system to be in restricted state, and it supports a reclaim of a specific library, rather than all libraries on the system.

Additionally this command provides an interface for the user to determine which catalogs have inconsistencies, and which libraries are affected.

If an auxiliary storage pool (ASP) group has been set for the current thread, this command will take affect for the independent ASPs in that ASP group and also for the system and basic user ASPs (ASP numbers 1-32).

This command should only be used when problems with the database cross-reference catalogs occur, and RCLSTG SELECT(*DBXREF) is not an option due to critical business requirements.

Because this command does not need to run in restricted state, it may not always be able to recover cross-reference information. In such cases, running RCLSTG SELECT(*DBXREF) will be required.

When the command is used to recover cross-reference information, the user must not allow applications to use or modify objects in a library that is being reclaimed. Failure to do this could cause unpredictable results.

If RCLDBXREF does not correct the problem, the library being reclaimed may have more inconsistencies than existed before using this command. If this happens, running RCLSTG SELECT(*DBXREF) will be required.

Restrictions:
• You must have all object (*ALLOBJ) special authority to use this command.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTION</td>
<td>Option</td>
<td>*CHECK, *FIX</td>
<td>Optional, Positional 1</td>
</tr>
<tr>
<td>LIB</td>
<td>Library</td>
<td>Name, *ERR</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Option (OPTION)

Specifies what action the command will take. The command can either check for problems, or attempt to fix them.

© Copyright IBM Corp. 1998, 2006
**CHECK**

The command should check to see if any cross-reference catalog inconsistencies have been found. As each catalog is inspected, a diagnostic message (CPD32A7) is sent to the job log for each library known to have inconsistent data in that catalog. After all of the cross-reference catalogs have been checked, an escape message (CPF32AC) is signaled if problems were found. If no inconsistencies are found in the catalogs, a completion message (CPC32AC) is sent indicating the cross-reference data appears to be correct.

**FIX**  
The command should attempt to fix the cross-reference data for the library specified for the `Library (LIB)` parameter.

**Note:** In some cases, the system may determine that a full reclaim of the database cross-reference catalogs is needed by running the Reclaim Storage (RCLSTG) command and specifying `SELECT(*DBXREF)`. In this case, a CPF32AB escape message is sent and the library’s cross-reference data will not be fixed.

When the `*FIX` option is specified, the user must be careful not to interrupt the reclaim process, or to attempt to use or modify objects in a library that is being reclaimed. Failure to do this could cause the reclaim to fail. If the reclaim fails, that library may have more catalog inconsistencies than existed before invoking this command. If this happens, it may be possible to recover by using the command again. If this does not correct the problem, a Reclaim Storage DBXREF (RCLSTG `SELECT(*DBXREF)`) will be required.

---

**Library (LIB)**

Specifies which library should be recovered.

**Note:** This parameter is only valid when `*FIX` is specified for the `Option (OPTION)` parameter.

**ERR**  
Recovery should be attempted for all libraries known to have had errors in the cross-reference catalogs. The command will attempt to recover all cross-reference data for any library found having inconsistent data when RCLDBXREF was run previously with `OPTION(*CHECK)` specified.

**name**  
Specify the name of the library for which cross-reference data will be fixed. All cross-reference data for the specified library will be recovered, even if the library is not known to have inconsistent data in the catalogs for it. The library does not need to exist to be specified on this command.

---

### Examples

**Example 1: Check for Cross-Reference Problems**

RCLDBXREF  OPTION(*CHECK)

This command checks the cross-reference catalogs for errors.

**Example 2: Recover the Cross-Reference Catalogs**

RCLDBXREF  OPTION(*FIX)  LIB(*ERR)

This command recovers the cross-reference information for all libraries known to be in error.

**Example 3: Recover the Cross-Reference Catalog Data for One Library**

RCLDBXREF  OPTION(*FIX)  LIB(ABCD)
This command recovers the cross-reference information for library ABCD.

Error messages

*ESCAPE Messages

CPF32AB
Database cross-reference information not recovered.

CPF32AC
Database cross-reference problem(s) exist.

CPF32A4
Internal failure in system cross-reference program.
Reclaim DDM Conversations (RCLDDMCNV)

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

The Reclaim Distributed Data Management Conversations (RCLDDMCNV) command reclaims all Distributed Data Management (DDM) source system conversations that are not currently being used by this source job, even if the attribute value for the job is *KEEP. By using the RCLDDMCNV command, you do not have to close all open files or do any of the other functions performed by the Reclaim Resources (RCLRSC) command. This command applies only to the DDM conversations for the job on the source system in which the command is entered.

Although this command applies to all DDM conversations used by this job, using it does not mean that all of them are taken down. A conversation is taken down only if there are no active users for that conversation.

There are no parameters for this command.

Parameters

None

Examples

RCLDDMCNV

This command checks all DDM conversations for the job in which the command is entered, determines if there are any users of each conversation, and reclaims each one not being used.

Error messages

None
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPT (Merge TCP/IP Host Table)
Reclaim Document Lib Object (RCLDLO)

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

The Reclaim Document Library Object (RCLDLO) command allows you to reclaim:
• a document
• a folder
• a folder and all documents and folders directly or indirectly contained within it
• internal Document Library system objects.
• internal Document Library system objects, unfiled distribution documents, and all filed folders and documents on the system. Related document details are synchronized.
• internal document library system objects and all filed folders and documents in one auxiliary storage pool (ASP). Related document details are synchronized.

When using the RCLDLO command to reclaim all document library objects, the objects are synchronized with the search details index and the text search index. The synchronization with the text search index is complete when all the requests to update the text search index generated using the RCLDLO command have been processed.

Restrictions:
• To reclaim internal document library system objects or all Document Library objects, you must have all object (*ALLOBJ) or security administrator (*SECADM) special authority. These objects can be reclaimed only when no folders or documents are in use.
• You must have exclusive use of the document or folder while it is being reclaimed. You do not need authority to a document or folder to reclaim it. You also do not need to be enrolled in the System Distribution Directory.
• While using this command, you may encounter an error message indicating that internal objects are locked. Another user may be using Document Library functions, which prevents the RCLDLO command from running. Retry this command after other Document Library activity has ended.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLR</td>
<td>In folder</td>
<td>Character value, *NONE</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>SYSOBJNAM</td>
<td>System object name</td>
<td>Name</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSOBJATR</td>
<td>System object attributes</td>
<td>*NONE, *INTDOC, *DST</td>
<td>Optional</td>
</tr>
<tr>
<td>ASP</td>
<td>Auxiliary storage pool ID</td>
<td>1-32, *ANY</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTFILE</td>
<td>File to receive output</td>
<td>Qualified object name</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: File to receive output</td>
<td>Name, *NONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 1998, 2006
### Document library object (DLO)

Specifies the name of object to reclaim.

This is a required parameter.

- **FLR**  
  A folder and all folders and documents directly or indirectly within it are to be reclaimed.

- **SYSOBJNAM**  
  A system object name is used to identify the folder or document to be reclaimed. This parameter must be used when reclaiming a document that is not in a folder including internal and distribution documents.

- **INT**  
  Internal document library system objects are to be reclaimed.

  **Note:** The internal document library system objects are used to manage the documents and folders on the system. RCLDLO DLO(*INT) is only necessary if the internal objects become damaged. If the internal objects are damaged, attempts to access documents and folders will result in the message CPF8A46 (Internal system objects are damaged), possibly followed by the message CPF9032 (Document interchange session not started).

- **DOCDTL**  
  Internal document library system objects and document details are to be reclaimed.

  DLO(*DOCDTL) synchronizes the relationships between all document library objects and their document details and will fix inconsistencies between them.

  **Note:** The RCLDLO DLO(*DOCDTL) command can be a long-running function, performing a subset of the RCLDLO DLO(*ALL) processing necessary to guarantee consistency between internal system objects, document details, and DLOs.

- **ALL**  
  Internal document library system objects and all documents and folders (as specified on the ASP parameter) are to be reclaimed. DLO(*ALL) synchronizes the relationships between all document library objects and their document details and can be used to fix inconsistencies between them.

  **Note:** The RCLDLO DLO(*ALL) command can be a long-running function, depending on the number of documents and folders on the system. If the RCLDLO command can be issued at the user's discretion, the user may wish to avoid the operation until the time required can be scheduled.

- **name**  
  Specify the name of the document or folder to be reclaimed.

### In folder (FLR)

Specifies the folder containing the folder or document to reclaim, or specifies the folder to reclaim along with all documents and folders directly or indirectly within it. A folder name is entered on this parameter only if a folder or document name or *FLR is entered on the Document library object (DLO) parameter.
*NONE
The folder or document to reclaim is not located in a folder.

name Specify the name of the folder that contains the document or folder to reclaim or specify the
name of the folder to reclaim along with all folders and documents directly contained within it.

System object name (SYSOBJNAM)
Specifies the system object name of the object to reclaim. A system object name must be entered on this
parameter if *SYSOBJNAM is specified on the Document library object (DLO) parameter.

System object attributes (SYSOBJATR)
Specifies the attributes of the object to reclaim. A value other than *NONE may be entered on this
parameter only if *SYSOBJNAM is specified on the Document library object (DLO) parameter.

*NONE
No attributes are specified for the object.

*INTDOC
The object to reclaim is an internal document.

*DST
The object to reclaim is a distribution document.

Auxiliary storage pool ID (ASP)
Specifies the identifier (ID) of the auxiliary storage pool (ASP) of the document library object to be
reclaimed. A value other than *ANY can be specified on this parameter only if *ALL or *DOCDTL is
specified on the DLO parameter.

*ANY
The objects to be reclaimed reside in any ASP. When *ALL is specified on the DLO parameter, all
document library objects on the system are reclaimed.

1-32 Only document library objects that reside in the specified ASP are to be reclaimed. All document
library objects in other ASPs are ignored. The value must designate an existing ASP that contains
document library objects. ASP 1 is the system ASP.

Note: Unfiled distribution documents are classified as document library objects in the system
ASP.

File to receive output (OUTFILE)
Specifies the name of the database file to which special output is directed. If the output file does not
exist, this command creates a database file in the specified library. If the file is created by this function,
the descriptive text is "OUTFILE created by RCLDLO" and the authority for users without specific
authority to the file is *EXCLUDE. A value other than *NONE can be specified on this parameter only if
*ALL or *DOCDTL is specified on the DLO parameter.
The output directed to this file includes the names of any documents that are physically damaged (and therefore unusable) or documents or folders that were missing from the system (and for which the document details have been removed). This file is intended to provide the user with a record of what was lost (such as a user ASP) when recovering from hardware failure.

**Qualifier 1: File to receive output**

*NONE  
No output is directed to a database file.

name  
Specify the qualified name of the database file that is to receive the output. This file can be reused when other RCLDLO commands are issued. Output is added to the file as specified on the OUTMBR parameter. The IBM-supplied database file, QSYS/QARCLDLO, cannot be specified.

**Qualifier 2: Library**

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

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

name  
Specify the name of the library to be searched.

---

**Output member options (OUTMBR)**

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

**Element 1: Member to receive output**

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

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

**Element 2: Replace or add records**

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

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

---

**Examples**

Example 1: Reclaiming a Folder

RCLDLO   DLO(FLR1)

This command reclaims folder FLR1.

Example 2: Reclaiming a Document Within a Folder
This command reclaims folder or document A in folder FLR2.

**Example 3: Reclaiming a Folder and All Documents and Folders Within It**

RCLDLO  DLO(*FLR)  FLR(FLR3)

This command reclaims folder FLR3 and all folders and documents directly or indirectly contained within it.

**Example 4: Reclaiming an Internal Document**

RCLDLO  DLO(*SYSOBJNAM)  SYSOBJNAM(AMBT133080)  SYSOBJATR(*INTDOC)

This command reclaims the internal document specified by the system object name AMBT133080.

**Example 5: Reclaiming a Distribution Document**

RCLDLO  DLO(*SYSOBJNAM)  SYSOBJNAM(AMBT133082)  SYSOBJATR(*DST)

This command reclaims the distribution document specified by the system object name AMBT133082.

**Example 6: Reclaiming Document Library System Objects**

RCLDLO  DLO(*INT)

This command reclaims internal document library system objects.

**Example 7: Reclaiming Document Library System Objects and Document Details**

RCLDLO  DLO(*DOCDTL)

This command reclaims internal document library system objects and document details for all folders and documents.

**Example 8: Reclaiming Document Library System Objects and All Documents and Folders**

RCLDLO  DLO(*ALL)

This command reclaims internal document library system objects and all documents and folders and synchronizes their document details.

**Example 9: Reclaiming Document Library System Objects and All Documents and Folders in an ASP**

RCLDLO  DLO(*ALL)  ASP(2)

This command reclaims internal document library system objects and all documents and folders residing in ASP 2 and synchronizes their document details.

---

**Error messages**

*ESCAPE Messages*

**CPF8A44**

Error occurred in document details.

**CPF8AA4**

Reclaim of &2 not successful.
CPF8A29

Reclaim of &2 partially successful.
Reclaim Library (RCLLIB)

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

The Reclaim Library (RCLLIB) command rebuilds the internal objects of a library that contain the object descriptive information for all objects in the library and the library object itself.

This command rebuilds, where possible, internal objects of the library that were damaged or destroyed.

Restrictions:
1. You must have object existence (*OBJEXIST) and use (*USE) authorities for the library to be rebuilt. This is the same authority required to delete a library with the Delete Library (DLTLIB) command.
2. Only the internal objects of a library which contain the object descriptive information are rebuilt. No other objects in the library are validated or rebuilt.
3. This command can be used on any library other than QTEMP.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIB</td>
<td>Library</td>
<td>Name</td>
<td>Required,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Positional 1</td>
</tr>
</tbody>
</table>

Library (LIB)

Specifies the library to be rebuilt.

This is a required parameter.

name Specify the name of the library to be rebuilt.

Examples

RCLLIB LIB(TESTLIB)

This command determines if the object descriptive information of library TESTLIB is damaged. The damaged parts of the library are rebuilt.

Error messages

*ESCAPE Messages

© Copyright IBM Corp. 1998, 2006
CPF210A
   Cannot reclaim library &1.

CPF210B
   Attempt to reclaim library &1 failed.

CPF2127
   User profile &2 damaged.

CPF9810
   Library &1 not found.

CPF9820
   Not authorized to use library &1.

CPF9830
   Cannot assign library &1.
Reclaim Object Links (RCLLNK)

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

The Reclaim Object Links (RCLLNK) command allows a single object or a group of objects to be reclaimed.

The RCLLNK command can be used to reclaim a directory tree where the directory, its contents, and the contents of all of its subdirectories are reclaimed. A subtree reclaim will attempt to reclaim as many objects as possible. A diagnostic message will be sent for each object that cannot be reclaimed. Additionally, an informational message will be sent if a specific problem is corrected, or a diagnostic message will be sent if a specific problem is unable to be corrected. If all of the objects have been reclaimed, with all of the problems corrected, then a completion message will be sent. Otherwise, an escape message will be sent.

The RCLLNK command does the following:
• Corrects object user profile problems
• Corrects user-defined file system problems
• Corrects internal object problems
• Removes invalid object links
• Handles damaged objects as specified on the Damaged object option (DMGOBJOPT) parameter
• Creates missing system objects

A full Reclaim Storage (RCLSTG) fixes the above problems, as well as others such as lost objects or problems which require the system to be in a restricted state. Unlike RCLSTG, the system does not have to be in a restricted state to run RCLLNK.

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:
• You must have all object (*ALLOBJ) special authority to run this command.
• Only objects found in the "root" (/), QOpenSys, or mounted user-defined file systems are eligible to be reclaimed. All other objects are ignored during a subtree reclaim.
• An independent auxiliary storage pool (ASP) must have a status of "Available" in order for objects residing on the independent ASP to be reclaimed.
• Storage freed objects are not reclaimed unless a storage freed object is specified on the Object (OBJ) parameter.
• Objects that are being saved or restored cannot be reclaimed.
• An object’s last access, data change, and attribute change date and time values are not normally updated as a result of being reclaimed. However, if the object has a problem that is corrected, these values may be updated.
• Problems found in the contents of a damaged directory object are not corrected.
• The directory conversion function must not be active while RCLLNK runs. The Convert Directory (CVTDIR) command with OPTION(*CHECK) can be used to verify directory conversion is not active.
### Objects

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ</td>
<td>Object</td>
<td><em>Path name</em></td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>SUBTREE</td>
<td>Directory subtree</td>
<td>*DIR, *NONE, *ALL</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>DMGOBJOPT</td>
<td>Damaged object option</td>
<td><em>Element list</em></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Usable objects</td>
<td>*KEEP, *DELETE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Unusable objects</td>
<td>*DELETE, *KEEP</td>
<td></td>
</tr>
</tbody>
</table>

#### Object (OBJ)

Specifies the path name of the object to be reclaimed. The object must be in the "root" (/), QOpenSys, or a user-defined file system. 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 cannot be specified ("*" or '?'). If the path name is qualified, it must be enclosed in apostrophes.

RCLLNK will not follow symbolic links for the last component in the path name.

If the last component in the path name is a block special file (*BLKSF) then only the block special file object will be reclaimed, not the user-defined file system that it represents.

The last component name in the path name cannot be ‘.’ (dot) or ‘..’ (dot-dot).

The effective root directory must be the "root" (/). Refer to the i5/OS PASE chroot command in the iSeries Information Center at [http://www.ibm.com/eserver/iseries/infocenter](http://www.ibm.com/eserver/iseries/infocenter) for more information.


#### Directory subtree (SUBTREE)

Specifies whether or not to reclaim the objects within the subtree if the object specified by the **Object (OBJ)** parameter is a directory.

- **DIR** The object specified by OBJ is reclaimed. If the object is a directory, its contents are reclaimed, but the contents of all of its subdirectories are not reclaimed.

- **NONE** Only the object specified by OBJ is reclaimed.

- **ALL** The object specified by OBJ is reclaimed. If the object is a directory, its contents, as well as the contents of all of its subdirectories, are reclaimed.
**Damaged object option (DMGOBJOPT)**

Specifies how damaged objects are handled that are considered usable or unusable. If no operations can be performed on a damaged object it is considered unusable. Otherwise, it is considered usable.

Deleting a usable damaged directory object will cause all its contents and the contents of all its subdirectories to become lost. These lost objects will need to be restored from media or can be recovered via a RCLSTG. The contents of an unusable damaged directory object are already lost.

The following considerations apply:
- Damaged file system root directory objects are not deleted
- Usable damaged objects that are in use are not deleted
- The contents of an unusable damaged directory object are not reclaimed
- The contents of a usable damaged directory object are reclaimed only if *KEEP is specified for usable objects

**Element 1: Usable objects**

*KEEP
Usable damaged objects are not deleted.

*DELETE
Usable damaged objects are deleted, if possible.

**Element 2: Unusable objects**

*DELETE
Unusable damaged objects are deleted, if possible.

*KEEP
Unusable damaged objects are not deleted.

**Examples**

**Example 1: Reclaim Object Links for a Directory**

RCLLNK OBJ('/MYOBJ') SUBTREE(*DIR)

The object MYOBJ will be reclaimed. If MYOBJ is a directory, all of the objects this directory contains will be reclaimed because *DIR is specified for the SUBTREE parameter.

**Example 2: Reclaim Object Links for an Object**

RCLLNK OBJ('/MYOBJ') SUBTREE(*NONE)

Only the object MYOBJ will be reclaimed because *NONE is specified for the SUBTREE parameter.

**Example 3: Reclaim Object Links for a Directory Subtree**

RCLLNK OBJ('/MYOBJ') SUBTREE(*ALL)

The object MYOBJ will be reclaimed. If MYOBJ is a directory, all of the objects this directory contains, as well as all of the objects contained in the subdirectories, will be reclaimed because *ALL is specified for the SUBTREE parameter.

**Example 4: Reclaim Damaged Objects in a Directory Subtree**

RCLLNK OBJ('/MYDIR') SUBTREE(*ALL) DMGOBJOPT(*KEEP *DELETE)
If the MYDIR directory is not damaged, keep all usable damaged objects and delete all unusable damaged objects found in the directory subtree of MYDIR. All other problems found are corrected, if necessary and possible.

Example 5: Search for All Damaged Objects in a Directory Subtree

RCLLNK OBJ('/MYOBJ') SUBTREE(*ALL) DMGOBJOPT(*KEEP *KEEP)

Check MYOBJ for damage. If MYOBJ is a directory, search for all damaged objects found in the directory subtree of MYOBJ. All other problems found are corrected, if necessary and possible.

Example 6: Delete All Damaged Objects in a Directory Subtree

RCLLNK OBJ('/MYOBJ') SUBTREE(*ALL) DMGOBJOPT(*DELETE *DELETE)

Delete MYOBJ if it is damaged. If MYOBJ is a directory that is not damaged, delete all damaged objects found in the directory subtree of MYOBJ. All other problems found are corrected, if necessary and possible.

Error messages

*ESCAPE Messages

CPF8206
Directory conversion cannot be active during RCLSTG or RCLLNK.

CPFA085
Home directory not found for user &1.

CPFA089
Pattern not allowed in path name.

CPFA0A2
Information passed to this operation was not valid.

CPFA0A7
Path name too long.

CPFA0A9
Object not found. Object is &1.

CPFA0B1

CPFA0DF
Error reclaiming objects in directories.

CPFA0F1
&1 of &2 object links reclaimed with &3 of &4 problems corrected.
Reclaim Objects by Owner (RCLOBJOWN)

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

The Reclaim Objects by Owner (RCLOBJOWN) command checks the objects owned by a user profile to make sure those objects which must be in a library are actually in one. If any of the owned objects are not in a library, they are inserted in one. If the object can only be in a specific system library, the object is inserted in that library. If the object can exist in any user library and is reclaimed from the system auxiliary storage pool (ASP 1) or a basic user ASP (ASPs 2-32), the object is inserted in library QRCL. If the object is reclaimed from a primary or secondary ASP, the object is inserted in library QRCLnnnnn where nnnnn is the number assigned by the operating system to the primary ASP.

When a user profile is deleted and the profile owns objects which are not in a library, message CPC2216 is sent after the user profile is deleted stating that the ownership for the objects was changed and that a reclaim operation, either RCLOBJOWN or RCLSTG, is necessary to place the objects in a library. Since RCLSTG can be a long-running command, you can try running RCLOBJOWN which is much faster because it only checks the objects owned by the specified user profile. If RCLOBJOWN is unable to reclaim the objects, you will need to run the RCLSTG command to reclaim the objects.

When a user profile is deleted because it is damaged, the objects owned by the user profile are left with no owner. In this case, you need to run the RCLSTG command to assign those objects to the QDFTOWN user profile. After running RCLSTG, you can use the Change Object Owner (CHGOBJOWN) command to change the ownership of objects assigned to QDFTOWN by the RCLSTG command.

If RCLOBJOWN is run against a user profile which has objects residing in an Independent Auxiliary Storage Pool (IASP), this IASP must be in AVAILABLE status, in order that the objects are reclaimed.

Note: The RCLOBJOWN command performs only a subset of the functions performed by the Reclaim Storage (RCLSTG) command. Refer to the RCLSTG command information for other reclaim commands that can be used to perform specific parts of the RCLSTG function.

Restrictions:
• This command is shipped with public *EXCLUDE authority.
• All subsystems must be inactive before the RCLOBJOWN command can be specified. The End System (ENDSYS) or End Subsystem (ENDSBS) command with *ALL specified on the SBS parameter can be used to make the subsystems inactive. You must have job control (*JOBCTL) authority to use the ENDSYS or the ENDSBS command.
• This command will not reclaim document (*DOC) objects or the folder (*FLR) objects where documents reside. It also will not reclaim directory (*DIR) objects, or the following types of objects which reside in directories:
  − stream file (*STMF)
  − symbolic link (*SYMLNK)
  − FIFO queue (*FIFO)
  − block special file (*BLKSF)
  − socket (*SOCKET)
User profile (USRPRF)

Specifies the user profile for which the reclaim will be performed.

**simple-name**

Specify the name of the user profile to be processed.

Examples

**Example 1: Reclaim Objects for a User Profile**

RCLOBJOWN      USRPRF(MYPROFILE)

This command will check all objects owned by MYPROFILE. If there are objects that are not in a library, they will be placed in a system library. A message is sent for each object inserted into a system library by the reclaim operation.

Error messages

***ESCAPE Messages**

CPF8204

Commitment control cannot be active during RCLSTG or RCLOBJOWN.

CPF8208

System not in proper state to reclaim objects by owner.

CPF8213

Object &1 cannot be inserted in &2 library.

CPF8215

User profile &1 damaged.

CPF8220

Library &1 damaged. Reclaim Objects by Owner (RCLOBJOWN) command ended.

CPF8232

Commitment definitions prevent RCLSTG or RCLOBJOWN.

CPF8282

User profile &1 not found.
Reclaim Optical (RCLOPT)

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

The Reclaim Optical (RCLOPT) command rebuilds the optical index database files and/or the internal library indexes. These files and indexes are used to maintain volume and directory information. Following are some of the reasons the files and indexes may need to be rebuilt: after CISC to RISC migration, after optical media is physical moved or removed by hand, after a hardware failure while media was being moved, or when a database or index has been damaged or destroyed. This command requires exclusive use of the directly-attached optical device.

Note: This command may take an extended period of time to complete - possibly more then 24 hours. The time to completion is influenced by several factors including: number of optical libraries, number of media in each library, capacity of each media, number of files and directories on each media and the options chosen on the command.

Restriction: You must have *USE authority to use this command. It is shipped with *EXCLUDE public authority.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLB</td>
<td>Optical media library</td>
<td>Name, *ALL</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>OPTION</td>
<td>Option</td>
<td>*SYNC, *UPDATE, *RESET</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>VOL</td>
<td>Volume identifier</td>
<td>Character value, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>DIR</td>
<td>Rebuild directory index</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
</tbody>
</table>

### Optical media library (MLB)

Specifies the directly-attached optical device for which the optical indexes are rebuilt.

*ALL  The optical indexes for all directly-attached optical devices are rebuilt.

*optical-device-name*

Specify the name of the directly-attached optical device for which the optical indexes are rebuilt.

### Option (OPTION)

Specifies which type of reclaim operation is performed.
**SYNC**

The optical volume index is synchronized with the internal device index. If a volume is not in the optical volume index, both the volume and the directory index entries are created for the volume. This option will usually complete in the shortest amount of time and, in most cases, should be attempted first before trying the *UPDATE or *RESET option.

**UPDATE**

The optical volume index is rebuilt and the optical directory index is optionally rebuilt with information read from the optical cartridge. In most cases this option will be used to rebuild the indexes for a single volume or optical cartridge.

**RESET**

The internal device index and the optical volume index are rebuilt. The optical directory index is optionally rebuilt with information read from the optical cartridge. Selecting *RESET requires every volume to be mounted into a drive at least once. The optical directory index is built if the Rebuild directory index parameter is set to *YES. Specifying DIR(*YES) will require every volume to be mounted a second time. In most cases *RESET should be used only after *SYNC and/or *UPDATE have already been attempted.

---

**Volume identifier (VOL)**

Specifies which volumes are used during the reclaim operation when OPTION(*UPDATE) is specified.

**ALL** All volumes in the optical device are used.

`volume-identifier`

Specify the volume identifier of a specific volume to use during the reclaim operation.

---

**Rebuild directory index (DIR)**

Specifies whether the optical directory index is rebuilt for each volume processed during Reclaim Optical processing. This parameter only applies to options *UPDATE and *RESET. When *SYNC is specified, DIR(*NO) will be used. In addition, this option does not apply to volumes which have a media format type of *UDF since directory index information is not maintained for *UDF volumes. When *UDF volumes are processed, DIR(*NO) will be used.

**NO** The optical directory index is not rebuilt for each volume processed.

**YES** The optical directory index is rebuilt for each volume processed. Only applies to *UPDATE and *RESET for non-UDF volumes.

**Note:** Choosing the value *NO may result in less time being required to complete the Reclaim Optical request.

---

**Examples**

```
RCLOPT   MLB(OPT01) OPTION(*UPDATE) VOL(VOL01) DIR(*YES)
```

This command re-creates both the optical volume index and the optical directory index for the optical volume VOL01 in the optical media library OPT01.
**Error messages**

*ESCAPE Messages*

**OPT0125**
Command &1 completed with errors, more information in job log.

**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.

**OPT1360**
Media directory corrupted on optical volume &1.

**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.

**OPT1605**
Media or device error occurred.

**OPT1640**
Error occurred reading files or directories.

**OPT1652**
Device &1 is not an optical media library.

**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.

OPT2040
   Error accessing backup control file.

OPT2155
   Reclaim Optical failed for optical device &1.

OPT2165
   Reclaim Optical failed for volume &1.

OPT2187
   Optical index inconsistency, reclaim optical required.

OPT2188
   Optical index files are damaged.

OPT2190
   Error occurred during reclaim optical processing while accessing volume &1.

OPT2191
   Error occurred while reading volume directory for volume &1.

OPT2301
   Internal system object in use.

OPT7740
   User not authorized to object &2 in library &3 type &4.
Reclaim Resources (RCLRSC)

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

The Reclaim Resources (RCLRSC) command is intended for use by the controlling program in an application to free resources that were allocated by programs that have finished running and returned control to the controlling program. The resources used by these programs can then be used by other programs running on the system. Based on the reference level specified by the LVL parameter, this command reclaims resources that are scoped to the default activation group. This command does not reclaim resources that are scoped to the job or resources that are scoped to any activation group other than the default activation group. The resources that are reclaimed by this command are:

- static storage
  - For an original program model (OPM) program, its static storage will be reclaimed. For a user state Integrated Language Environment (ILE) program, its static storage will be marked so that it will be re-initialized on the next call. Static storage is not affected for a service program, a system state ILE program, or any ILE program running in an activation group other than the default activation group.
- open files of object type *FILE
- user interface manager (UIM) application resources
- Common Programming Interface (CPI) Communications conversations
- hierarchical file systems (HFS) resources
- user-defined communications sessions.

In addition, after distributed data management (DDM) files are closed based on the criteria described above, all DDM source system conversations in this job that are not currently in use will be ended, regardless of the activation group or reference level from which they were started.

The RCLRSC command is not needed to reclaim the files and static storage of most programs, such as CL programs that end (return) normally, RPG programs that have the last record (LR) indicator set on, and COBOL programs. The RCLRSC command should not be used if it might be processed while any COBOL program is still active in the application.

For more information on the appropriate use of this command, see the documentation associated with the programming language(s) used in the application.

Restrictions:
1. This command is not thread safe. However, it can be run in the primary thread of a multi-threaded job.
2. Do not specify LVL(*CALLER) on this command if it is used in a CL program that also uses the Send File (SNDF), Receive File (RCVF), Send/Receive File (SNDRCVF) commands. Specifying RCLRSC LVL(*CALLER) in such a program causes unpredictable results when the SNDF, RCVF, or SNDRCVF commands are used after the program runs.
3. Do not specify LVL(*CALLER) on this command if it is issued from a command line, as this will result in unpredictable results.
Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVL</td>
<td>Program level</td>
<td>* , *CALLER</td>
<td>Optional, Positional 1</td>
</tr>
<tr>
<td>OPTION</td>
<td>Close option</td>
<td>*NORMAL, *ABNORMAL</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**Program level (LVL)**

Specifies the reference level at which resources are reclaimed.

* The reference level is the program or procedure that contains this RCLRSC command. The resources are reclaimed for programs or procedures that have finished running and returned control to this program.

*CALLER  
The reference level is the program or procedure that called the program or procedure containing this RCLRSC command. This value allows controlling programs or procedures written in a high-level language to call a CL program to reclaim resources to the level of the controlling program or procedure. The effect is the same as if the command were issued from the controlling program or procedure.

*Note:* Using the *CALLER value can cause unexpected results when running the RCLRSC command from a command line or from within a program that works with open files.

**Close option (OPTION)**

Specifies whether a normal or abnormal close notification is sent to the attached host system when mixed, communications, binary synchronous (BSC), and intersystem communications function (ICF) files are closed. This parameter is ignored for all other files and objects.

*NORMAL  
The attached host system is given a normal close notification when mixed, communications, BSC, and ICF files are closed.

*ABNORMAL  
The attached host system is given an abnormal close notification when mixed, communications, BSC, and ICF files are closed. Use this when the controlling program detects error conditions that should be communicated to the host systems (the error condition need not be file-related).

**Examples**

**Example 1 (OPM)**

```
PROGA:
   CALL PROGB
   RCLRSC:
   CALL PROGC
   RCLRSC:
```

IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPTH (Merge TCP/IP Host Table)
In this example, PROGA is a controlling program in an application. PROGA calls other programs, which return control to PROGA when they have finished running. Because control is returned to the next sequential instruction, the RCLRSC command is issued following each CALL command to free the static storage that was used by the called program, and to close the files that were left open.

**Example 2 (OPM)**

```
PROGA
| CALL PROGB
| RCLRSC

PROGD
| CALL LVL(+CALLER)
| RETURN
```

In this example, PROGA is a controlling program that is written in a high-level language. The RCLRSC command cannot be issued from the high-level language program so PROGD, a CL program, is called to issue the command. When the RCLRSC command is issued in PROGD, the static storage used by PROGB and PROGC is freed; files that were left open are closed.

**Example 3 (OPM)**

```
PROGA
| CALL PROGB
| RCLRSC

PROGB
| CALL PROGC
| RETURN

PROGC
```

In this example, PROGA is a controlling program. When the RCLRSC command is issued, the static storage used by PROGB and PROGC is freed; files that were left open are closed.

**Example 4 (OPM)**

```
PROGA
| CALL PROGB
| TFRCTL PROGC

PROGC
| CALL PROGB
| RCLRSC
```

In this example, PROGA calls PROGB and, after returning from PROGB, PROGA transfers to program PROGC. Because PROGB has already been called, static storage exists, and the call to PROGB from PROGC does not cause any new allocation for static storage; PROGC cannot reclaim the static storage.
used by PROGB. If PROGB opened files when it was called by PROGA, these files would remain open; if PROGB opened files when it was called by PROGC, these files are closed.

**Example 5 (OPM)**

In this example, PROGA calls PROGB, which in turn calls PROGC. PROGC opens a file. Control returns to PROGA. PROGA calls PROGB a second time, and PROGB invokes RCLRSC. Since RCLRSC is based upon the position in the stack of the current invocation of a program, the file opened by PROGC is not affected. PROGC was invoked earlier than the current PROGB. In order for RCLRSC to close the file, the RCLRSC command will have to be invoked by PROGA.

**Example 6 (OPM & ILE)**

This example shows how ILE procedures and activation groups are affected by the RCLRSC command.

In this example, PROGA is a program running in the default activation group. PROGA calls program PROGB which runs in the default activation group. PROGB calls ILE procedure PROCC which runs in the default activation group. PROCC calls ILE procedure PROCD which causes activation group AG1 to be created. PROCD returns to PROCC. PROCC returns to PROGB. PROGB returns to PROGA, which then calls the RCLRSC command.

PROGA calls the RCLRSC command. Any resources in use by PROGA are still open, since PROGA is still in use. Any resources by program PROGB or procedure PROCC are reclaimed, since the program and procedure ran in the default activation group and are no longer active. Any resources opened by procedure PROCD are left alone, since procedure PROCD ran in activation group AG1 and only the default activation group is affected by the RCLRSC command.

Any other use of the RCLRSC command can result in files remaining open and storage remaining allocated.
Error messages

*ESCAPE Messages

CPF180C
  Function &1 not allowed.

CPF1892
  Function &1 not allowed.
Reclaim Spool Storage (RCLSPLSTG)

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

The Reclaim Spool Storage (RCLSPLSTG) command reclaims unused storage for spooled files that have not been used for more than the number of days specified by the user. Spooled files are stored with database file members on the system. When a spooled file is deleted, the member is emptied but not deleted. Therefore, the member can be reused for the next spool file created. Reusing empty members improves the performance time when creating new spooled files. The RCLSPLSTG command deletes unused and empty database members in the system auxiliary storage pool (ASP 1) and any basic user auxiliary storage pools (ASPs 2-32). The RCLSPLSTG command will not delete unused and empty database members in primary or secondary auxiliary storage pools (ASPs). The QRCLSPLSTG system value can be set to automatically delete unused and empty database members in primary or secondary ASPs. This command uses synchronous processing. More information about synchronous processing is in the Backup and Recovery book, SC41-5304.

Restriction: To use this command, you must be signed on as QPGMR, QSYSOPR, QSRV, or QSRVBAS, or have *ALLOBJ authority.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAYS</td>
<td>Days</td>
<td>1-366, *NONE</td>
<td>Required, Positional 1</td>
</tr>
</tbody>
</table>

Days (DAYS)

Specifies an interval in days. If existing spool storage has not been used to create new spooled files in the number of days specified, it will be deleted.

This is a required parameter.

*NONE

No interval is used. All unused spool storage is deleted. No storage will remain for creating new spooled files. This will lengthen the time it takes to create a new spooled file.

Note: Using this value can have adverse effects on system performance. More information is in the Files and file systems topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

1-366 Specify the number of days. The specified number of days is measured to the second.
Examples
RCLSPLOSTG DAYS (30)

This command recovers all unused storage for spooled files that have remained unused for more than 30 days. When storage has been unused for 1 second over 30 days it is reclaimed because a date and time stamp is placed on the storage area.

Error messages
None
Reclaim Storage (RCLSTG)

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

The Reclaim Storage (RCLSTG) command corrects, where possible, objects that were incompletely updated (such as database files, libraries, device descriptions, directories and stream files) and user profiles containing incorrectly recorded object ownership information. Any unusable objects or fragments are deleted.

This command reclaims all objects secured by an authorization list that is damaged or destroyed and assigns the objects to the authorization list QRCLAUTL.

Because the amount of time required to run this command varies with the number of objects in auxiliary storage, the system will display a panel to the work station where the command was specified, showing the progress of the command’s execution. The ‘Time Remaining’ column will show blanks for a RCLSTG step if no statistics have been collected yet that would enable the RCLSTG command to estimate the total time required for that step.

You can specify *YES for the ESTIMATE parameter to get an estimate of how long the RCLSTG command will take to run. When ESTIMATE(*YES) is specified, messages that show the estimated amount of time are sent to the job log, and no reclaim function is performed.

The RCLSTG command can also be used to reclaim storage when, during an IPL, not enough storage is available to make the system fully operational. In that case, the system operator can specify the command immediately after receiving the message about insufficient storage.

If very little additional auxiliary storage is available, the system overhead required to run the RCLSTG command may need more than the remaining storage; in that case, the RCLSTG command fails.

Note: The RCLSTG command can be a long-running function, depending on the number and type of objects in the system, and the amount of damage to them. Because RCLSTG touches each object multiple times, having enough memory can significantly reduce the time required to run RCLSTG. Conversely, having too little memory can lead to storage thrashing which can significantly increase the time required to run RCLSTG. If database file objects are damaged, the keyed access paths may need to be rebuilt; that operation takes a substantial amount of time. If the RCLSTG command can be run at the user’s discretion, the user may want to avoid the operation until the required time can be scheduled.

You can select to run just the directory recovery portion of RCLSTG by specifying SELECT(*DIR). This will reclaim only directories and objects related to the integrated file system.

There are several reclaim commands that perform a subset of the RCLSTG’s functions. These commands are: Reclaim Object Links (RCLLNK), Reclaim Database Cross-Reference (RCLDBXREF), Reclaim Objects by Owner (RCLOBJOWN), Reclaim Document Lib Object (RCLDLO), Reclaim Library (RCLLIB) and Reclaim Spool Storage (RCLSPLSTG). Refer to those commands for details on the functions that they perform.

Restrictions:
1. This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.
2. If option *SYSBAS is specified on the ASPDEV parameter, all subsystems must be inactive before the RCLSTG command can be specified. The End System (ENDSYS) or End Subsystem (ENDSBS)

© Copyright IBM Corp. 1998, 2006
command with *ALL specified on the SBS parameter can be used to make the subsystems inactive. You must have job control (*JOBCTL) authority to use the ENDSYS or the ENDSBS command.

3. Only permanent objects in auxiliary storage are reclaimed; temporary objects are reclaimed by running a system initial program load (IPL).

4. Before running the RCLSTG command after an IPL, you may need to wait several minutes for the IPL to complete. Use the Work with Active Jobs (WRKACTJOB) command to verify that no jobs are running.

5. If option *SYSBAS is specified for the ASPDEV parameter, this job must be in the controlling subsystem and must be the only job active in the system.

6. Network server descriptions (NWSD) must be varied-off in order to run RCLSTG.

7. When option *SYSBAS is specified for the ASPDEV parameter, all the auxiliary storage pool (ASP) devices configured in the system must be in the VARY OFF status.

8. If an auxiliary storage pool (ASP) device or an ASP group is specified on the ASPDEV parameter, it cannot have active users or be in use by a different job. The ASP device or group has to be in the AVAILABLE or FAILED status to allow the RCLSTG command to run.

9. The directory conversion function must not be active while RCLSTG runs. The Convert Directory (CVTDIR) command with OPTION(*CHECK) can be used to verify directory conversion is not active. Otherwise, option OMIT(*DIR) should be specified to omit the directory recovery portion of the reclaim function.

10. In order to provide an estimate of the amount of time that a future RCLSTG will require, statistics are collected when running the RCLSTG ESTIMATE(*NO) command. If you specify ESTIMATE(*YES) and these statistics are not available, escape message CPF8281 is issued.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESTIMATE</td>
<td>Estimate time required</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>SELECT</td>
<td>Select</td>
<td>*ALL, *DBXREF, *DIR</td>
<td>Optional</td>
</tr>
<tr>
<td>OMIT</td>
<td>Omit</td>
<td>*NONE, *DBXREF, *DIR</td>
<td>Optional</td>
</tr>
<tr>
<td>ASPDEV</td>
<td>ASP device or group</td>
<td>Name, *SYSBAS</td>
<td>Optional</td>
</tr>
</tbody>
</table>

### Estimate time required (ESTIMATE)

Specifies whether to calculate an estimate for the amount of time that RCLSTG command will take to run. The estimate is calculated by using statistics collected during previous RCLSTG operations and the values specified for the other RCLSTG parameters.

*NO  The estimate function is not performed. The reclaim function specified by the other parameters is performed.

*YES  The estimate function is performed for the type of reclaim specified by the other parameters. Nothing is reclaimed. If SELECT(*DBXREF) is specified, no estimate can be provided. Instead, the time it took to run the last RCLSTG SELECT(*DBXREF) is provided, if available. Messages are displayed showing the results of the estimate.
Select (SELECT)

Specifies whether to perform all reclaim functions or only one specific reclaim function.

*ALL  All reclaim functions are performed including, but not limited to, database cross-reference table reclaim and directory recovery functions.

*DBXREF Only the database cross-reference table reclaim function is performed.

*DIR  Only the directory recovery portion of the reclaim function is performed.

Omit (OMIT)

Specifies the reclaim functions to be omitted from the reclaim operation.

*NONE  No reclaim functions are omitted.

*DBXREF The database cross-reference table reclaim function is omitted.

*DIR  The directory recovery portion of the reclaim function is omitted.

ASP device or group (ASPDEV)

Specifies the auxiliary storage pool (ASP) to be reclaimed.

*SYSBAS  The system ASP and all basic ASPs are reclaimed. The system ASP has an ASP number of 1. Basic ASPs have ASP numbers of 2 through 32.

auxiliary-storage-pool-device-name  The specified ASP device is reclaimed. ASP devices have ASP numbers greater than 32. Reclaim storage for an ASP device can be run without being in restricted state. The ASP device must be in the AVAILABLE or FAILED status to reclaim it. You can submit multiple jobs, each performing RCLSTG on a different ASP device, to reclaim multiple ASP devices in parallel.

auxiliary-storage-pool-group-name  The primary ASP and the secondary ASPs within the ASP group are reclaimed. The name of any ASP within the ASP group is accepted. Primary and secondary ASPs have ASP numbers greater than 32. Reclaim storage for an ASP group can be run without being in restricted state. The ASP device must be in the AVAILABLE or FAILED status to reclaim it. You can submit multiple jobs, each performing RCLSTG on a different ASP group, to reclaim multiple ASP groups in parallel.

Examples

Example 1: Reclaim Storage of the System ASP and All Basic ASPs

RCLSTG

This command, specified interactively, locates all system objects. Objects without owners are given default owners, and those that are lost from their specified libraries are inserted into the QRCL library or the default library, or are deleted.
Objects that are lost from their specified directories are inserted into the ‘/QReclaim’ directory (if the object was originally located in the root file system) or the ‘/QOpenSys/QReclaim’ directory (if the object was originally located in the QOpenSys file system).

Lost objects that are deleted are certain user objects and certain i5/OS system objects that are damaged and not usable.

The QRCL library, which is created (when needed) by the RCLSTG command, is a permanent library.

The ‘/QReclaim’ and ‘/QOpenSys/QReclaim’ directories, which are created (when needed) by the RCLSTG command, are permanent directories; but if they contain no objects at the end of the operation because they were all reclaimed, the directories are deleted.

Example 2: Reclaim Storage to Reclaim the Database Cross-reference Table
RCLSTG SELECT(+DBXREF)

This command reclaims the database cross-reference table.

Example 3: Reclaim Storage of the Entire System that Omit’s the Reclaim of the Database Cross-reference Table
RCLSTG OMIT(+DBXREF)

This command performs all reclaim storage functions but omits the reclaim of the database cross-reference table.

Example 4: Reclaim Storage that Omits the Reclaim of the Directories
RCLSTG OMIT(+DIR)

This command performs all reclaim storage functions but omits the reclaim of the directories.

Example 5: Reclaim Storage of an ASP Device
RCLSTG ASPDEV(MYASPDEV)

This command reclaims storage for the ASP device name MYASPDEV.

Example 6: Reclaim Storage to Reclaim the Directory Portion of the System ASP and All Basic ASPs
RCLSTG SELECT(+DIR)

Objects that are lost from their specified directories are inserted into the ‘/QReclaim’ directory (if the object was originally located in the root file system) or the ‘/QOpenSys/QReclaim’ directory (if the object was originally located in the QOpenSys file system).

Example 7: Reclaim Storage to Reclaim the Directories of an ASP Device
RCLSTG SELECT(+DIR) ASPDEV(MYASPDEV)

This command reclaims directories for the ASP device name MYASPDEV.

Example 8: Estimate the Time that the Next Full RCLSTG Will Take to Run
RCLSTG ESTIMATE(+YES)

This command estimates the time that the next full RCLSTG will take to run for each one of the major steps:
• Reading objects from disk.
• File ID table recovery.
• Directory recovery.
• Object description verification.

Messages are sent to the job log giving the estimated amount of time for each major RCLSTG step.

**Example 9: Estimate the Time that the Next RCLSTG of an ASP Device Will Take to Run**

```
RCLSTG  ESTIMATE(*YES)  ASPDEV(MYASPDEV)
```

This command estimates the time that the next RCLSTG of ASP device name MYASPDEV will take. Messages are sent to the job log giving the estimated amount of time for each major RCLSTG step.

---

**Error messages**

*ESCAPE Messages*

**CPF180B**
Function &1 not allowed.

**CPF180C**
Function &1 not allowed.

**CPF2119**
Library &1 locked.

**CPF2120**
Library &1 locked.

**CPF2126**
&1 libraries not validated.

**CPF2127**
User profile &2 damaged.

**CPF5729**
Not able to allocate object &1.

**CPF8201**
User profile &1 does not exist or is damaged.

**CPF8204**
Commitment control cannot be active during RCLSTG or RCLOBJOWN.

**CPF8205**
Library &1 does not exist or is damaged.

**CPF8206**
Directory conversion cannot be active during RCLSTG or RCLLNK.

**CPF8209**
System not in proper state to reclaim storage.

**CPF8210**
Reclaim storage failed for auxiliary storage pool device or group &2.

**CPF8211**
Library &1 damaged. RCLSTG command ended.

**CPF8212**
SELECT(*DBXREF) or OMIT(*DBXREF) not allowed when reclaiming a UDFS ASP.
CPF8214
    Estimate option cannot be performed.

CPF8216
    SELECT("DBXREF") or OMIT("DBXREF") not allowed for a UDFS ASP.

CPF8224
    Duplicate object found while moving or renaming member.

CPF8232
    Commitment definitions prevent RCLSTG or RCLOBJOWN.

CPF8251
    RCLSTG command ended. Library &1 damaged.

CPF8252
    Error occurred during rebuild of damaged library &1.

CPF8281
    Unable to estimate time required to run RCLSTG.

CPF9814
    Device &1 not found.

CPFA473
    Network server &1 must be varied off.

CPF88ED
    Device description &1 not correct for operation.
Reclaim Temporary Storage (RCLTMPSTG)

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

The Reclaim Temporary Storage (RCLTMPSTG) command allows you to reclaim storage used by temporarily decompressed copies of panel groups, menus, display files, and printer files, thereby freeing up system storage space.

- Compressed Objects are objects that consume less storage space than decompressed objects. When a compressed object is used or a compressed program is called, a decompressed version of the object automatically becomes available to the user.
- Decompressed Objects are objects that use the system storage space allocated to them and are in a final, ready-to-use state.
- Temporarily Decompressed Objects are temporarily decompressed copies of compressed objects. The system allocates storage space for the decompressed objects, which is consumed by the temporary copies until the system or the user determines that the temporary storage space needs to be reclaimed.

Temporary storage is automatically reclaimed when:
- the RCLTMPSTG command is run
- the next IPL is run
- the object is used often enough to cause the system to permanently decompress it

When an object is permanently decompressed, the compressed version of the object is destroyed as well as any temporary forms of the object; however, compressed versions remain intact as long as the objects are temporarily decompressed.

Restrictions:
1. This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.
2. The user must have object management authority to the object specified and execute authority to the library.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAYS</td>
<td>Days unused</td>
<td>1-366, 7, *NONE</td>
<td>Optional, Positional 2</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 1998, 2006
Library (LIB)

Specifies the name of the library from which storage is reclaimed. All temporarily decompressed objects that are in the library you specify on this parameter and that have not been used for more than the number of days you specify on the Days unused prompt (DAYS parameter) are reclaimed.

The possible library values are:

*ALL All libraries in the system, including QSYS, are searched.
*LIBL All libraries in the library list for the current thread are searched until the first match is found.
*USRLIBL If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched.
*CURLIB Only the current library is searched. If no current library entry exists in the library list, QGPL is used.
*ALLUSR All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

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

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

QDSNX QRCLxxxxx QUSRJJS QUSRVxRxMx
QGPL QSRVAGT QUSRINFSKR
QGPL3B QSYS2 QUSRNOTES
QMTC QSYS2xxxxx QUSROND
QMTC2 QS36F QUSRPOSgs
QMPGDATA QUSER3B QUSRPOSSA
QMQMDBATA QUSRADS9 QUSRQMSVR
QMPMPQOC QUSRBRM QUSRDRARS
QPFRODATA QUSRDR1CL QUSRYS
QRC QUSRDR1DB QUSRV

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

library-name

Specify the name of the library to be searched.

Days unused (DAYS)

Specifies the number of days an object has not been used or changed. If a temporarily decompressed object has not been used or changed for more than the specified number of days, it is reclaimed. If it has been used or changed, it is left temporarily decompressed.

The possible values are:

7 Objects that have not been used or changed for more than seven days are reclaimed.
*NONE
   The object is reclaimed regardless of the number of days it has not been used or changed.

*days-unused
   Specify the number of days. Valid values range from 1 through 366.

Examples
RCLTMPSTG    LIB(QGPL)

This command reclaims the space consumed by all of the temporarily decompressed copies of objects in library QGPL that have not been used or changed in the last 7 days.

Error messages

*ESCAPE Messages

CPF2110
   Library &1 not found.

CPF2113
   Cannot allocate library &1.

CPF2176
   Library &1 damaged.

CPF3B07
   &1KB storage reclaimed, &5 objects not processed.

CPF9838
   User profile storage limit exceeded.
The Receive Distribution (RCVDST) command allows you to receive incoming distributions such as documents or files. The documents or files can be placed in folders or document objects or can be placed in an output file for processing.

**Restriction:** The requester of the command must be enrolled in the system distribution directory. If you request distribution information for another user, you must have been given permission to work on behalf of that user with the Grant User Permission (GRTUSRPMN) command. Personal distribution cannot be requested if the requester is working on behalf of another user.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSTID</td>
<td>Distribution identifier</td>
<td>Character value</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>USRID</td>
<td>User identifier</td>
<td>Single values: *CURRENT</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: User ID</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Address</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td>DOC</td>
<td>Document</td>
<td>Character value, *NONE</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>FLR</td>
<td>In folder</td>
<td>Character value, *NONE</td>
<td>Optional, Positional 3</td>
</tr>
<tr>
<td>OUTFILE</td>
<td>File to receive output</td>
<td>Single values: *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values: Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: File to receive output</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>OUTMBR</td>
<td>Output member options</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Member to receive output</td>
<td>Name, *FIRST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Replace or add records</td>
<td>*REPLACE, *ADD</td>
<td></td>
</tr>
<tr>
<td>OUTDTATYP</td>
<td>Type of data for output</td>
<td>Single values: *DFT, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>ACKRCV</td>
<td>Acknowledge receipt</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>DSTIDEXN</td>
<td>Distribution ID extension</td>
<td>1-99, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>KEEF</td>
<td>Keep in mail log</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Distribution identifier (DSTID)

Specifies the unique distribution identifier of the distribution. The identifier is assigned to the distribution by the system that originated it. Only incoming distributions can be received. If the identifier represents an outgoing distribution, an error message is returned.

distribution-id

The distribution identifier is composed of the second part of the sender’s user ID (padded on the right to 8 characters), the first part of the sender’s user ID (padded on the right to 8 characters), and a 4-digit zoned sequence number with the leading zeros. For example, ‘NEWYORK SMITH 0204’. This parameter is required when ‘DSTID is specified on the Information to be sent prompt (TYPE parameter).

This is a required parameter.

User identifier (USRID)

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

*CURRENT

You are performing the request for yourself.

user-ID

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

user-ID-address

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

Document (DOC)

Specifies the name of the document object in which the distribution is placed when it is received. This document must not already exist and it is created as a private document. The document is either owned by you, or by an authority-granting user for whom you work.

*NONE

The distribution being received is not placed in a document object.

document-name

Specify the name of the document in which the distribution is placed. A maximum of 12 characters can be specified.
In folder (FLR)

Specifies the name of the folder that contains the document receiving the distribution. The folder must already exist and you must have the authority to create new documents in the folder.

*NONE

The document being received is not placed in a folder. Specify this value if the document is received into a database file for processing and Document (DOC parameter) is not specified.

folder-name

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

File to receive output (OUTFILE)

Specifies the name and library of the database file to which the output is directed. If the output file does not exist, this command creates a database file in the specified library. The authority for users with no specific authority is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library’s create authority.

*NONE

The output is not directed to a database file.

data-base-file-name

Specify the name of the database file that receives the output.

The possible library values are:

*LIBL

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

*CURLIB

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

library-name

Specify the library where the database file is located.

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

Output member options (OUTMBR)

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

The possible member to receive output values are:

*FIRST

The first member in the file receives the output. If the member does not exist, the system creates a member with the name of the file specified on the File to receive output prompt (OUTFILE parameter).
Specify the name of the file member that receives the output. If the member does not exist, the system creates the file member. If the member already exists, the system adds records to the end of the member or clears the member and then adds the records.

The possible add or replace values are:

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

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

**Type of data for output (OUTDTATYP)**

Specifies which type of distribution data is written to the database file.

*DFT
The following record codes are written to the output file:

<table>
<thead>
<tr>
<th>Record code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>010</td>
<td>Distribution Description</td>
</tr>
<tr>
<td>020</td>
<td>Message Text</td>
</tr>
<tr>
<td>105</td>
<td>Document Description</td>
</tr>
<tr>
<td>800</td>
<td>Document Data</td>
</tr>
</tbody>
</table>

*ALL
All record formats are written to the output file.

*DSTINFO
The distribution description record is written. The record code is 010.

*MSG
The message text record is written. The record code is 020.

*DOCD
The document description record is written. The record code is 105.

*DOCCLS
The document class record is written. The record code is 155.

*SUBJECT
The subject records are written. The record code is 165.

*FILCAB
The file cabinet reference record is written. The record code is 160.

*AUTHOR
The author records are written. The record code is 145.

*KWD
The keyword records are written. The record code is 170.

*CPYLST
The copy list records are written. The record code is 150.
*FILDATE
The file date record is written. The record code is 125.

*EXPDATE
The expiration date record is written. The record code is 115.

*DOCDATE
The document date record is written. The record code is 120.

*CRTDATE
The create date record is written. The record code is 110.

*ACTDATE
The action due date record is written. The record code is 135.

*CHGDATE
The date last changed record is written. The record code is 130.

*CMPDATE
The completion date record is written. The record code is 140.

*REF
The reference record is written. The record code is 175.

*STATUS
The status record is written. The record code is 180.

*PROJECT
The project record is written. The record code is 185.

*AUTUSR
The authorizing Userid and Address is written. This is the user that authorized the content of this distribution. The record code is 190.

*DSTEXPDATE
The distribution expiration date and time is written. The record code is 195.

*R PYDATE
The reply requested date and time is written. The record code is 200.

*IDP
The interchange document profile (IDP) is written. The record code is 500.

*DOC
The document data records are written. The record code is 800.

---

**Acknowledge receipt (ACKRCV)**

Specifies whether a confirmation of delivery (acknowledgement) is sent back to the sender of the distribution.

*YES
The confirmation of delivery (COD) is sent back to the sender.

*NO
The confirmation of delivery is not sent back to the sender.

---

**Distribution ID extension (DSTIDEXN)**

Specifies the extension of the distribution identifier (if any) specified on the Distribution identifier prompt (DSTID parameter). This 2-digit extension has a value ranging from 01 through 99 that uniquely identifies duplicate distributions. The default value is 01.
There is no duplicate distribution. *NONE is equivalent to an extension of 01.

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

---

**Keep in mail log (KEEP)**

Specifies whether the received distribution is deleted from the mail log or kept in the mail log.

*NO* When all the information requested has been written to the OUTFILE or DOC, the distribution is removed from the user's incoming mail.

*YES* When all the information requested has been written to the OUTFILE or to DOC, the distribution is not removed from the user's incoming mail.

---

**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.

If the values specified on the Distribution identifier prompt (DSTID parameter) and User identifier prompt (USRID parameter) are being read from an output file created by the Query Distribution (QRYDST) command, specify '930 500' on this parameter.

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

*DEVD* 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.

---

**Examples**

**Example 1: Receiving Current User Distribution**

RCVDST DISTID('SYSTEM1 USERA 0001')
OUTFILE(MYLIB/MYFILE) OUTMBR(MYMBR *ADD)
OUTDTATYP(*ALL) CMDCHRID(*DEVD)
This command receives the current user distribution into output file MYFILE located in library MYLIB. The distribution is added to member MYMBR. All output file information is added to the output file MYFILE.

**Example 2: Receiving Distribution Sent to a User**

RCVDST DSTID('BAKER RCH3BP 0019') DSTINDEXN(01)
OUTFILE(JOWL/DOCUMENTS) USRID(+CURRENT)

This command receives a distribution that was sent to a user. It is copied into the first member in a database file called DOCUMENTS in a library called JOWL.

---

**Error messages**

*ESCAPE Messages*

CPF8A87
Document name &2 not correct.

CPF8A97
Folder name &1 not correct.

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.

CPF9096
Cannot use CMDCHRID(*DEVD), DOCCHRID(*DEVD) in batch job.

CPF9098
Distribution not received.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

CPF9847
Error occurred while closing file &1 in library &2.

CPF9860
Error occurred during output file processing.
Receive File (RCVF)

Where allowed to run:
- Batch program (*BFGM)
- Interactive program (*IPGM)

Threadsafe: Conditional

The Receive File (RCVF) command is used by a CL procedure to receive data from a display device or database file. The command reads a record from the file and puts the data from the record into one or more CL variables. These CL variables were automatically declared in the program when the CL source program was compiled and a Declare File (DCLF) command was processed as part of the source. There is one CL variable for each field in the record format used to receive the data. The data that is entered by a user at the display or is contained in the input record is copied into CL variables in the program by the RCVF command, where it is processed by the program.

Only one record format, of those specified in the DCLF command, can be specified in each RCVF command. If the file has not been opened by a previous RCVF, SNDRCVF, or SNDF command, it is opened by this command. If the file has been previously closed due to an end-of-file condition on a previous RCVF command, an error occurs. The file specified in this command can be overridden if the override command is entered before the file is opened. If the file specified in the DCLF command was a display file when the program was compiled, the file may only be overridden to another display file. If the file was a database file, the file may only be overridden to another database file that has a single record format. However, care should be taken that the fields in the overriding record format correspond to the CL variables declared in the program.

Restrictions:
- This command is valid only within a CL procedure.
- This command is conditionally threadsafe. The RCVF command is threadsafe when issued against a database file. RCVF is not threadsafe when issued against a display file, and should not be used in a job with multiple threads to receive data from a display file.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV</td>
<td>Display device</td>
<td>Name, *FILE</td>
<td>Optional, Positional 1</td>
</tr>
<tr>
<td>RCFMT</td>
<td>Record format</td>
<td>Name, *FILE</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>OPNID</td>
<td>Open file identifier</td>
<td>Simple name, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>WAIT</td>
<td>Wait</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Display device (DEV)

Specifies the name of the display device from which data is to be received. If a CL variable name is used in this parameter, only one RCVF command is needed in the program to receive data from several devices. (The variable specifying the device name can be changed while repeatedly running the same command.) This parameter may be specified only if the file is a display device file.

*FILE The user’s data is to be received from the device associated with the device file (the device file that was declared in the FILE parameter of the DCLF command). If more than one device name is specified in the device file, *FILE cannot be specified.

name Specify the name of the device or the name of the CL variable that contains the name of the device from which the user’s data is to be received.

Record format (RCDFMT)

Specifies the name of the record format that is used to receive data from the file. The format contains all the fields in the record. This parameter must be coded with a record format name if there is more than one record format in the device file. If the file is a database file, the specified record format is used to map the data from the record into the CL variables. The actual record format name in the file at run time may be different. RCVF ignores the INVITE DDS keyword.

*FILE There is only one record format in the device file; that is the format in which the data is to be received. If more than one record format is specified in the device file, *FILE cannot be specified.

name Specify the name of the record format in which the data records from the display device are to be received. A CL variable cannot be used to specify the record format name.

Open file identifier (OPNID)

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

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

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

Wait (WAIT)

Specifies whether the CL procedure waits for the data to be received from the user’s device or continues processing the commands that follow this RCVF command. If WAIT(*NO) is specified, the program must issue a WAIT command later in the program to complete the input operation. This parameter may be specified only if the file is a display device file.

*YES The program waits until the input operation from the device is completed; the next command is not processed until then.
*NO  The program does not wait for the input data; commands continue running until a WAIT command is reached later in the program.

---

## Examples

### Example 1: Receive Data from Database File

```cl
DCLF FILE(MENU1) :
RCVF OPNID(*NONE)
```

The CL procedure receives data from the database file named MENU1. The RCVF command is associated with the Declare File command that does not have an open file identifier specified.

### Example 2: Receive Data from Display Device

```cl
DCLF FILE(SCREENX) RCDFMT(R1 R2) :
RCVF DEV(DISPLAY2) RCDFMT(R1)
```

The CL procedure receives data from the user at the display station named DISPLAY2. The data is received in the record format named R1 in the device file named SCREENX. The procedure waits for the user data before it continues processing.

### Example 3: Handling End-of-File Exception

```cl
DCLF FILE(INPUT) OPNID(INFILE1) :
RCVF OPNID(INFILE1)
MONMSG CPF0864 EXEC(GOTO EOF)
```

The CL procedure receives a record sequentially from the database file named INPUT. The procedure monitors for the end-of-file exception CPF0864 and goes to label EOF when the message is received.

### Example 4: Using RCVF with WAIT command

```cl
DCLF FILE(MSCREEN) RCDFMT(MIN1 MIN2 MIN3) :
RCVF DEV(&DNAME) RCDFMT(MIN2) WAIT(*NO)
WAIT DEV(&DNAME)
```

The CL procedure receives user data from several devices one at a time by way of the device file named MSCREEN. The procedure receives data from the device named in the variable &DNAME using the record format MIN2, but it does not wait for the data to come in. The same RCVF command is used to receive data from several devices; because the CL variable &DNAME is used, only the device name in the DEV parameter must be changed each time the command is run. A WAIT command for each device must be issued later in the procedure because the WAIT command actually receives the data. Both the RCVF and the WAIT commands may be processed for each device (one at a time) to send data to the procedure. If a user response is delayed, the commands can be processed as many times as necessary until the user responds with the data or a End Receive (ENDRCV) command cancels the request.

---

## Error messages

**ESCAPE Messages**
CPF0859
File override caused I/O buffer size to be exceeded.

CPF0860
File &1 in &2 not a data base file.

CPF0861
File &1 in library &2 is not a display file.

CPF0863
Value of binary data too large for decimal CL variable.

CPF0864
End of file detected for file &1 in &2.

CPF0865
File &1 has more than one record format.

CPF0883
*FILE not valid in DEV parameter for file &1.

CPF0886
Record contains a data field that is not valid.

CPF4101
File &2 in library &3 not found or inline data file missing.

CPF502A
Variable length record error on member &4.

CPF502B
Error occurred in trigger program.

CPF502D
Referential constraint violation on member &4.

CPF502E
Referential constraints could not be validated for member &4.

CPF502F
Check constraint violation on member &4.

CPF5029
Data mapping error on member &4.

CPF503A
Referential constraint violation on member &4.

CPF503B
Record could not be inserted or updated in member &4.

CPF5068
Program device &4 not found in file &2 in library &3.

CPF5070
File &2 in library &3 has no program devices acquired.
Receive Journal Entry (RCVJRNE)

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

The Receive Journal Entry (RCVJRNE) command allows a specified user exit program to continuously receive journal entries. This program can be set up, for example, to write the entries either (1) to an ICF file, supplying updates to a file on a backup system, or (2) on a tape, imitating a journal-to-tape function. The information in the journal entries received can be used to update the objects being journaled to minimize the loss of data in the event of a disk failure, and to update objects on a backup system in case of a system failure on the primary system.

The value specified on the ENTFMT parameter determines the format of the journal entries passed to the exit program.

Restrictions:

- If the sequence number is reset in the range of the receivers specified, the first occurrence of FROMENTLRG or FROMENT is used, if they are specified. If TOENTLRG or TOENT is specified, the first occurrence after the FROMENTLRG or FROMENT entry is used, if FROMENTLRG or FROMENT is specified. Otherwise the first occurrence is used.
- The FILE, OBJ, OBJPATH, OBJFID, SUBTREE, PATTERN, JRNCDE, ENTTYP, JOB, PGM, USRPRF, CCIDLrg, CMTCYCID, and DEPENT parameters can be used to specify a subset of all available entries within a range of journal entries.
  - If no values are specified using these parameters, all available journal entries are received.
  - If more than one of these parameters are specified, then a journal entry must satisfy all of the values specified on these parameters, except when *IGNFILSLT or *IGNOBJSLT is specified on the JRNCDE parameter.
  - If a journal code is specified on the JRNCDE parameter and *IGNFILSLT is the second element of that journal code, then journal entries with the specified journal code are selected if they satisfy all selection criteria except what is specified on the FILE parameter.
  - If a journal code is specified on the JRNCDE parameter and *IGNOBJSLT is the second element of that journal code, then journal entries with the specified journal code are selected if they satisfy all selection criteria except what is specified on the OBJ, OBJPATH, OBJFID, SUBTREE, and PATTERN parameters.
- The JOB, PGM, and USRPRF parameters cannot be used to specify selection criteria if one or more journal receivers in the specified receiver range was attached to the journal when a receiver size option (RCVSIZOPT) or a fixed length data option (FIXLENDTA) that would have omitted this data was in effect.
- If more than the maximum number of objects is identified (32767 objects), an error occurs and no entries are received. This restriction is ignored if *ALLFILE is specified or no objects are specified.
- When journal caching is being used, entries that are in the cache are not returned.
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Keyword</strong></td>
<td><strong>Description</strong></td>
<td><strong>Choices</strong></td>
<td><strong>Notes</strong></td>
</tr>
<tr>
<td>JRN</td>
<td>Journal</td>
<td>Qualified object name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Journal</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>EXITPGM</td>
<td>Program to receive entries</td>
<td>Qualified object name</td>
<td>Required, Positional 2</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Program to receive entries</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>FILE</td>
<td>Journaled physical file</td>
<td>Single values: *ALLFILE Other values (up to 300 repetitions): Element list</td>
<td>Optional, Positional 3</td>
</tr>
<tr>
<td></td>
<td>Element 1: File</td>
<td>Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: File</td>
<td>Name, *ALL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Member</td>
<td>Name, *FIRST, *ALL, *NONE</td>
<td></td>
</tr>
<tr>
<td>OBJ</td>
<td>Objects</td>
<td>Values (up to 300 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Object</td>
<td>Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Object</td>
<td>Name, *ALL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Object type</td>
<td>*FILE, *DTAARA, *DTAQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 3: Member, if data base file</td>
<td>Name, *FIRST, *ALL, *NONE</td>
<td></td>
</tr>
<tr>
<td>OBJPATH</td>
<td>Objects</td>
<td>Values (up to 300 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Name</td>
<td>Path name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Include or omit</td>
<td>*INCLUDE, *OMIT</td>
<td></td>
</tr>
<tr>
<td>OBJFID</td>
<td>File identifier</td>
<td>Values (up to 300 repetitions): Hexadecimal value</td>
<td>Optional</td>
</tr>
<tr>
<td>SUBTREE</td>
<td>Directory subtree</td>
<td>*NONE, *ALL</td>
<td>Optional</td>
</tr>
<tr>
<td>PATTERN</td>
<td>Name pattern</td>
<td>Values (up to 20 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Pattern</td>
<td>Character value, *</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Include or omit</td>
<td>*INCLUDE, *OMIT</td>
<td></td>
</tr>
<tr>
<td>RCVRNG</td>
<td>Range of journal receivers</td>
<td>Single values: *CURRENT, *CURCHAIN Other values: Element list</td>
<td>Optional, Positional 4</td>
</tr>
<tr>
<td></td>
<td>Element 1: Starting journal receiver</td>
<td>Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Starting journal receiver</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Ending journal receiver</td>
<td>Single values: *CURRENT Other values: Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Ending journal receiver</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>FROMENTLRG</td>
<td>Starting large sequence number</td>
<td>Character value, *FIRST</td>
<td>Optional</td>
</tr>
<tr>
<td>FROMTIME</td>
<td>Starting date and time</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Starting date</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Starting time</td>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>TOENTLRG</td>
<td>Ending large sequence number</td>
<td>Character value, *LAST, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Choices</td>
<td>Notes</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>TOTIME</strong></td>
<td>Ending date and time</td>
<td><em>Element list</em></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Ending date</td>
<td><em>Date</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Ending time</td>
<td><em>Time</em></td>
<td></td>
</tr>
<tr>
<td><strong>NBREN</strong></td>
<td>Number of journal entries</td>
<td><em>Integer</em>, <em>ALL</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>JRNCDE</strong></td>
<td>Journal codes</td>
<td>*Single values: *ALL, <em>CTL</em> *Other values (up to 16 repetitions): <em>Element list</em></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Journal code value</td>
<td><em>A, B, C, D, E, F, J, L, M, P, R, Q, S, T, U</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Journal code selection</td>
<td>*ALLSLT, *IGNFILSLT, *IGNOBJSLT</td>
<td></td>
</tr>
<tr>
<td><strong>ENTTYP</strong></td>
<td>Journal entry types</td>
<td>*Single values: *ALL, <em>RCD</em> *Other values (up to 300 repetitions): <em>Character value</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>JOB</strong></td>
<td>Job name</td>
<td>*Single values: *ALL, * *Other values: <em>Qualified job name</em></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Job name</td>
<td><em>Name</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: User</td>
<td><em>Name</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 3: Number</td>
<td><em>000000-999999</em></td>
<td></td>
</tr>
<tr>
<td><strong>PGM</strong></td>
<td>Program</td>
<td><em>Name</em>, <em>ALL</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>USRPRF</strong></td>
<td>User profile</td>
<td><em>Name</em>, <em>ALL</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>CCIDLRC</strong></td>
<td>Commit cycle large identifier</td>
<td><em>Character value</em>, <em>ALL</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>DEPENT</strong></td>
<td>Dependent entries</td>
<td>*ALL, <em>NONE</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>FMTMINDTA</strong></td>
<td>Format minimized data</td>
<td>*NO, <em>YES</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>NULLINDLEN</strong></td>
<td>Null value indicators length</td>
<td>*Single values: <em>ENTFMT</em> *Other values: <em>Element list</em></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Field data format</td>
<td><em>1-8000</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Variable length field length</td>
<td><em>1-8000</em></td>
<td></td>
</tr>
<tr>
<td><strong>DELAY</strong></td>
<td>Delay time</td>
<td><em>Element list</em></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Delay time value</td>
<td>1-99999, *30, *NEXTENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Maximum delay time value</td>
<td>1-99999, <em>CLS</em></td>
<td></td>
</tr>
<tr>
<td><strong>BLKLEN</strong></td>
<td>Block length</td>
<td><em>32-4000</em>, *NONE, <em>CALC</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>JRNENTFMT</strong></td>
<td>Journal entry format</td>
<td><em>RJNE0100, RJNE0200</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>RTNPR</strong></td>
<td>Return pointers</td>
<td>*NONE, *SYSMNG, <em>USRMNG</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>INCENT</strong></td>
<td>Include entries</td>
<td>*CONFIRMED, <em>ALL</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>FROMENT</strong></td>
<td>Starting sequence number</td>
<td>1-999999999999, <em>FIRST</em></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>TOENT</strong></td>
<td>Ending sequence number</td>
<td>1-999999999999, *LAST, *NONE</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>CMTCYCID</strong></td>
<td>Commit cycle identifier</td>
<td>1-999999999999, <em>ALL</em></td>
<td>Optional</td>
</tr>
</tbody>
</table>
Journal (JRN)

Specifies the journal from which the journal entries are received.

This is a required parameter.

Qualifier 1: Journal

\textit{journal-name}

Specify the name of the journal.

Qualifier 2: Library

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

\textbf{*CURLIB}\quad The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

\textit{library-name}\quad Specify the name of the library to be searched.

Program to receive entries (EXITPGM)

Specifies a user-written exit program that controls the receiving of each journal entry passed from the command. Additional information on the interface between this command and the exit program is supplied after the listing of possible values for this parameter, and is described in more detail in the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

This is a required parameter.

Qualifier 1: Program to receive entries

\textit{program-name}\quad Specify the name of the exit program that controls the reception of each journal entry passed from the command.

Qualifier 2: Library

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

\textbf{*CURLIB}\quad The current library for the job is used to locate the program. If no library is specified as the current library for the job, QGPL is used.

\textit{library-name}\quad Specify the library where the program is located.

Additional Information on the Exit Program Interface

When the program is called, two parameters are passed to it at a time. A single journal entry or a block of journal entries is passed in the first parameter.

- If a single journal entry is passed, and if the length of the parameter defined by the program is smaller than the length of the journal entry, the journal entry passed to the program is truncated. If the length of the parameter defined by the program is greater than the length of the journal entry, the parameter...
positions beyond the length of the journal entry contain nonessential information. The user’s program should not specifically refer to data in the positions beyond the length of the journal entry.

- If BLKLEN("NONE) was specified, then the exit program may indicate to the system that multiple entries should be returned in subsequent calls to the exit program by specifying 8 in the first byte of the second parameter. This is called “Block Mode” and the exit program must specify the size of the block in bytes as a zoned value in the first 5 bytes of the first parameter. If an error is made in this specification, only one journal entry is passed in the block. If BLKLEN("NONE) was not specified, then specifying 8 in the first byte of the second parameter will have no effect and the first 5 bytes of the first parameter will be ignored.

- If pointers are being returned, this first parameter must be aligned on a 16-byte boundary since journal entry specific data could include actual pointers.

For *TYPE1, *TYPE2, *TYPE3 and *TYPE4 formats there is a zoned journal entry length field that is filled with zeros at the end of the single journal entry or block of journal entries passed. This field indicates that the last journal entry has been passed. The format of the information in each journal entry is shown in the ENTFMT parameter description. The format of the first parameter is detailed in the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

**Note:** The maximum length of the parameter specification in the exit program is language dependent (for example, for CL, the maximum length is 9999). For more information about limitations refer to the corresponding programming language book.

A character variable of LEN(3) is passed in the second exit program parameter. This parameter will be passed from the system to the exit program and can be passed from the exit program to the system. Its values are presented in the following lists:

- Information in the first byte of the second parameter:

  **Char(1)**
  
  Passed to the Exit Program
  
  0  No journal entry is passed on this call of the exit program.
  1  A single journal entry is passed to the exit program.
  2  A block of one or more journal entries is passed to the exit program.
  3  No journal entry is passed on this call to the exit program, and no more entries can be passed, because the journal receiver that was attached when the receive journal entry operation started is no longer attached.

  **Note:** The system ends the RCVJRNE command after calling the exit program once with a reason code of 3.

  4  No journal entry is passed on this call to the exit program, and no more entries can be passed unless the remote journal is activated.

  **Note:** This value can only be passed to the exit program when receiving journal entries from the attached receiver of a remote journal and the journal state for the journal is currently *INACTIVE.

  **Char(1)**
  
  Passed to the System from the Exit Program
  
  8  Requests the command processing program to start passing one or more journal entries in a block. If BLKLEN("NONE) was not specified, then specifying 8 in the first byte of the second parameter will have no effect and the first 5 bytes of the first parameter will be ignored.

  9  Requests the RCVJRNE command to end. The exit program returns control to the system.
Char(1)

Passed to the Exit Program

N  Additional journal entries are not currently available to be passed after this call of the exit program, or the RCVJRNE command will be ending after this call of the exit program.

Y  Additional journal entries are currently available to be passed after this call of the exit program.

Any information passed from the exit program to the system in this second character will be ignored.

This second byte of the second exit program parameter is provided whether journal entries are being passed as a single journal entry per call of the exit program, or as a block of journal entries per call.

• Information in the third byte of the second parameter:

Char(1)

Passed to the Exit Program

'00'x  One or more journal entries are being passed to the exit program and the object names in the fixed length portion of each journal entry do not necessarily reflect the name of the object at the time the journal entry was deposited into the journal.

**Note:** This value is returned only when receiving journal entries from a journal receiver that was attached to a journal prior to V4R2M0.

0  No journal entries are currently being passed, so the information normally returned in this byte is not applicable.

1  One or more journal entries are being passed to the exit program and the object names in the fixed length portion of each journal entry reflect the name of the object at the time the journal entry was deposited into the journal.

2  One or more journal entries are being passed to the exit program and the object names in the fixed length portion of each journal entry do not necessarily reflect the name of the object at the time the journal entry was deposited into the journal. The object name in the fixed length portion of the journal entry may be returned as a previously known name for the object prior to the journal entry being deposited into the journal or be returned as *UNKNOWN.

**Note:** This value will only be returned when receiving journal entries from a remote journal and the remote journal is currently being caught up from its source journal. A remote journal is being caught up from its source journal when the Change Remote Journal (CHGRMTJRN) command or Change Journal State (QjoChangeJournalState) API is invoked and is currently replicating journal entries to the remote journal. After the call to the CHGRMTJRN command or QjoChangeJournalState API returns, the remote journal is maintained with a synchronous or asynchronous delivery mode, and the remote journal is no longer being caught up.

Any information passed from the exit program to the system in the third byte will be ignored. The second byte of the second exit program parameter is provided whether journal entries are being passed as a single journal entry per call of the exit program, or as a block of journal entries per call.

**Note:** When an N is passed to the exit program in the second byte of the second parameter indicating that no additional journal entries are currently available, it does not necessarily mean that when the exit program returns, that the RCVJRNE command will have to wait for additional journal entries to be deposited into the journal. By the time the exit program returns, additional journal entries may already be available and depending upon what was specified on the DELAY parameter, may or may not be immediately passed to the exit program. If DELAY(N) was specified the system will wait N seconds before passing the journal entries to the exit program. If DELAY(*NEXTENT) was specified, the journal entries will immediately be passed to the exit program.

The third byte of the second exit program parameter is provided whether journal entries are being processed as a single journal entry per call of the exit program, or as a block of journal entries per call. When returned for a block of journal entries, the attribute applies to the object names for all of the journal entries being returned in the block.
For more information on the exit program and these two parameters used to receive the journal entries, see the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Journaled physical file (FILE)

Specifies a maximum of 300 qualified file names whose journal entries are received. This parameter also specifies the name of the file member whose journal entries are to be received.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, or OBJFID) may be specified, but not both.

To determine which journal entries are to be received, based on the specified file member name, the following is done:

- If the journal is a local journal, and if the specified file member currently exists on the system, the journal identifier is determined from the specified file member. All journal entries in the specified receiver range for that journal identifier are received.
- If the journal is a remote journal, or if the specified file member does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified file member. All journal entries in the specified receiver range for those journal identifiers are received. Specify the library name or *CURLIB to have entries returned for the file.

There may be more than one journal identifier associated with a specified object within the specified receiver range. This can happen when a journaled object is deleted, and then a new object is created with the same name and journaled to the same journal.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved, or restored. See the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information.

2. When specifying a database file on this parameter, journal entries with the following journal code values are received only if they satisfy the values specified on the other parameters:
   - Journal code D (database file-level information entries).
   - Journal code F (file member-level information entries).
   - Journal code R (record-level information entries).
   - Journal code U (user-generated entries).
   - Other journal codes, if *IGNFILSLT is specified on that journal code. If *ALLSLT is specified on that journal code, no journal entries with that code are received.

Single values

*ALLFILE

The search for the journal entries received is not limited to a specified file name. All journal entries are received, regardless of which objects, if any, the entries are associated with.

Element 1: Journaled physical file

Qualifier 1: Journaled physical file

*ALL  Journal entries for all physical files in the specified library (the library name must be specified)

whose journaled changes are currently in the journal receiver are received. If *ALL is specified and the user does not have the required authority to all of the files, an error occurs, and the command ends.

**physical-file-name**
Specify the name of the database physical file for which a journal entry is received.

**Qualifier 2: Library**

*LIBL  All libraries in the library list for the current thread are searched until the first match is found.
*CURLIB  The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

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

**Element 2: Member**

*FIRST  Entries for the database file and the first member in the file are received.
*ALL  Entries for the database file and all the currently existing members of the file are received.
*NONE  Only entries for the database file are received. Entries for members of the file are not received.

**member-name**
Specify the name of the member for which journal entries are received.

If *ALL is specified for the file-name element, this member name is used for all applicable files in the library. For example, if library-name/*ALL *FIRST is specified on the FILE parameter, the journal entries of the first members of all applicable files in the specified library are received.

**Objects (OBJ)**

Specifies a maximum of 300 qualified object names whose journal entries are to be received. The possible object types are *FILE, *DTAARA, and *DTAQ. If *FILE is specified, this parameter also specifies the name of the file member whose journal entries are to be received.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, or OBJFID) may be specified, but not both.

To determine which journal entries are to be received, based on the specified object name, the following is done:

- If the journal is a local journal, and if the specified object currently exists on the system, the journal identifier is determined from the specified object. All journal entries in the specified receiver range for that journal identifier are received.
- If the journal is a remote journal, or if the specified object does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified object. All journal entries in the specified receiver range for those journal identifiers are received. Specify the library name or *CURLIB to have entries returned for an object.

There may be more than one journal identifier associated with a specified object within the specified receiver range. This can happen when a journaled object is deleted, and then a new object is created with the same name and journaled to the same journal.
Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information.

2. When specifying an object on this parameter, journal entries with the following journal code values are received only if they satisfy the values specified on the other parameters in addition to the object name specification:
   - Journal code D (database file-level information entries).
   - Journal code E (data area information entries).
   - Journal code F (file member-level information entries).
   - Journal code Q (data queue information entries).
   - Journal code R (record-level information entries).
   - Journal code U (user-generated entries).
   - Other journal codes, if *IGNOBJSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are received.

Element 1: Object

Qualifier 1: Object

*ALL  Journal entries for all objects of the specified object type in the specified library (the library name must be specified) whose journaled changes are currently in the journal receiver are received. The library name must be specified. If *ALL is specified and the user does not have the required authority for all objects in the library, a message is sent and the command ends.

object-name
   Specify the name of the object whose journaled changes are to be received.

Qualifier 2: Library

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

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

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

Element 2: Object type

Specify the object type of the object whose journaled changes are to be received.

*FILE  Entries for database files and database file members are received.

*DTAARA
   Entries for data areas are received.

*DTAQ
   Entries for data queues are received.

Element 3: Member, if data base file

Specify the name of the member in the file whose journal entries are to be received. If *ALL is specified for the first part of this parameter, the value specified for the member name is used for all applicable files in the library. For example, if *FIRST is specified, the journal entries of the first member of all applicable files in the specified library are received.
Note: If the specified object type is not *FILE, the member name element value is ignored.

*FIRST
Entries for the database file and the first member in the file are received.

*ALL Entries for the database file and all the currently existing members of the file are received.

*NONE
Only entries for the database file are received. Entries for members of the file are not received.

member-name
Specify the name of the member for which journal entries are received.

If *ALL is specified for the object-name element, this member name is used for all applicable files in the library. For example, if library-name/*ALL *FILE *FIRST is specified on the OBJ parameter, the journal entries of the first members of all applicable files in the specified library are received.

Objects (OBJPATH)

Specifies a maximum of 300 objects whose journal entries are to be received. Only objects whose path name identifies an object of type *STMF, *DIR or *SYMLNK that are in the "root" ('/'), QOpenSys, and user-defined file systems are supported. All other objects are ignored.

This parameter is not valid for remote journals.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, or OBJFID) may be specified, but not both.

Only objects that are currently linked with the specified path name and have a journal identifier associated with them are used in journal entry selection. If the specified object does exist, the journal identifier associated with that link is used for journal entry selection. If a specified object does not exist or does not have a journal identifier associated with it, that link is not used in selecting journal entries and no error is sent.

Notes:
1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information.
2. When specifying an object on this parameter, journal entries with the following journal code values are received only if they satisfy the values specified on the other parameters in addition to the object name specification:
   - Journal code B (integrated file system information entries).
   - Journal code U (user-generated entries).
   - Other journal codes, if *IGNOBJSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are received.

Element 1: Name

path-name
Entries for objects identified by the path name are received.

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

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

Element 2: Include or omit

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

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

*INCLUDE

The objects that match the object name pattern are to be included in determining what journal entries are received unless overridden by an *OMIT specification.

*OMIT

The objects that match the object name pattern are not to be included in determining what journal entries are received. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

File identifier (OBJFID)

Specifies a maximum of 300 file identifiers (FID) whose journal entries are to be received. FIDs are a unique identifier associated with integrated file system related objects. This field is input in hexadecimal format. Only objects whose FID identifies an object of type *STMF, *DIR or *SYMLNK that are in the "root" ("/"), QOpenSys, and user-defined file systems are supported. All other objects are ignored.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, or OBJFID) may be specified, but not both.

To determine which journal entries are to be received, based on the specified file identifier, the following is done:

• If the journal is a local journal, and if the specified object currently exists on the system, the journal identifier is determined from the specified object. All journal entries in the specified receiver range for that journal identifier are received.

• If the journal is a remote journal, or if the specified object does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified object. All journal entries in the specified receiver range for those journal identifiers are received.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information.

2. When specifying an object on this parameter, journal entries with the following journal code values are received only if they satisfy the values specified on the other parameters in addition to the object name specification:

• Journal code B (integrated file system information entries).
• Journal code U (user-generated entries).
• Other journal codes, if *IGNOBJSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are received.

_file-identifier_
Entries for objects identified with the FID are received.

**Directory subtree (SUBTREE)**
Specifies whether the directory subtrees are included in determining the objects for which journal entries are to be received.

**Note:** This parameter is only valid if one or more path names were specified on the OBJPATH parameter.

*NONE
Only the objects that match the selection criteria are processed. The objects within selected directories are not implicitly processed.

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

**Name pattern (PATTERN)**
Specifies a maximum of 20 patterns to be used to include or omit objects for which journal entries are to be received.

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

If the Name Pattern parameter is not specified the default will be to match all patterns.

**Note:** This parameter is only valid if one or more path names were specified on the OBJPATH parameter.

**Element 1: Pattern**

* All objects that match the input OBJPATH parameter are to be included.

_name-pattern_
Specify the pattern to be used to include or omit objects for which journal entries are received. Only the last part of the path name will be considered for the name pattern match. Path name delimiters are not allowed in the name pattern.

If the Name Pattern parameter is not specified the default will be to match all patterns.

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

**Element 2: Include or omit**
The second element specifies whether names that match the pattern should be included or omitted from the operation. Note that in determining whether a name matches a pattern, relative name patterns are always treated as relative to the current working directory.

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

*INCLUDE  
The objects that match the object name pattern are included in the operation, unless overridden by an *OMIT specification.

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

---

**Range of journal receivers (RCVRNG)**

Specifies the starting (first) and ending (last) journal receivers used in the search for the journal entries that are received. The system starts the search with the starting journal receiver (as specified by the first value) and proceeds through the receiver chain until the ending journal receiver (as specified by the last value) is processed.

If a problem is found in the receiver chain (such as damaged or not-found receivers) before the search operation begins, the system tries to use the second of the dual receivers. If these receivers also are damaged or not found, the operation ends.

**Single values**

*CURRENT  
The journal receiver that is currently attached when starting to receive journal entries is used.

*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.

**Element 1: Starting journal receiver**

**Qualifier 1: Starting journal receiver**

`starting-journal-receiver-name`  
Specify the name of the first journal receiver containing journal entries that are to be received.

**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**
Single values

*CURRENT
The journal receiver that is currently attached when starting to receive journal entries is used.

Qualifier 1: Starting journal receiver

ending-journal-receiver-name
Specify the name of the last journal receiver containing journal entries that can be received. If the end of the receiver chain is reached before a receiver with this name is found, an error message is sent and no journal entries are received.

Note: The maximum number of receivers in the range is 1024. If more receivers than this maximum are specified, an exception is signaled, and no journal entries are received.

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 first journal entry considered for reception.

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

*FIRST  The first journal entry in the specified journal receiver range is the first entry considered for reception.

starting-sequence-number
Specify the sequence number of the first journal entry considered for reception. The possible range is 1 to 18,446,744,073,709,551,600.

Starting date and time (FROMTIME)

Specifies the date and time of the first journal entry considered for reception. The starting date and time of the first journal entry created either at or after the specified starting date and time is the starting point for reception of the journal entries.

Element 1: Starting date

starting-date
Specify a starting date.

Element 2: Starting time
starting-time
Specify a starting time. The time can be specified in 24-hour format with or without a time separator:

- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.

---

**Ending large sequence number (TOENTLRG)**

Specifies the last journal entry considered for reception.

**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.

*NONE
No journal entry is specified. Journal entries are passed to the exit program until the command is canceled (by a cancel request or a cancel job command) or until an end reason code (9) is set by the exit program. If there are no more entries to pass, the RCVJRNE command waits the number of seconds indicated on the DELAY parameter before trying to find more entries to pass.

**Note:** TOENTLRG(*NONE) is valid only if the RCVRNG parameter specifies a receiver that is currently attached when starting to receive journal entries.

*LAST
The last journal entry in the journal receiver range specified is the last entry considered for reception.

**ending-sequence-number**
Specify the sequence number of the final journal entry considered for reception. The possible range is 1 to 18,446,744,073,709,551,600.

**Note:** The values specified for the from and to prompts can be the same. For example, FROMENTLRG(234) and TOENTLRG(234) can be specified.

---

**Ending date and time (TOTIME)**

Specifies the date and time of the last journal entry considered for reception. The ending date and time of the journal entry created at or before the specified ending date and time is the ending point for reception of the journal entries.

**Element 1: Ending date**

**ending-date**
Specify the date of the last entry received.

**Element 2: Ending time**

**ending-time**
Specify the creation time of the last entry received. The time can be specified in 24-hour format with or without a time separator:
• Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
• With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Number of journal entries (NBRENT)
Specifies the total number of journal entries that are received.

*ALL All journal entries included in the specified journal receiver range that satisfy the selection values are received.

value Specify the maximum number of journal entries be received. If the specified journal entry identified by the TOENTLRG, TOENT, or TOTIME parameter is reached before the value specified for NBRENT is met, the command ends normally.

Journal codes (JRNCDE)
Specifies the journal code that is used to limit the entries being considered for reception.

Single values

*ALL The journal entries received are not limited to those containing a specified code.

*CTL The journal entries received are those written to control the journal functions. These journal entries have codes J or F.

Element 1: Journal code value

journal-code
Specify the journal code to which journal entries are limited. Only journal entries with the specified journal code are received.

An explanation of the journal codes that can be specified is in the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Element 2: Journal code selection

*ALLSLT The journal entries with the specified journal code are received only if all other selection parameters are satisfied.

*IGNFILSLT Journal entries having the specified journal code are received only if all selection parameters, except the FILE parameter, are satisfied.

Note: This value is not valid for journal codes D, F, and R. This value is not valid if the OBJ, OBJPATH, or OBJFID parameters are specified.

*IGNOBJSLT Journal entries having the specified journal code are received only if all selection parameters are satisfied except OBJ, OBJPATH, OBJFID, SUBTREE, and PATTERN.
Note: This value is not valid for journal codes B, D, E, F, Q, and R. This value is not valid if the FILE parameter is specified.

---

**Journal entry types (ENTTYP)**

Specifies whether to limit the journal entries received to those of a specified journal entry type.

**Single values**

- **ALL** The journal entries that can be received are not limited to those of a specified journal entry type.
- **RCD** Only entries that have an entry type for record level operations are received. The following entry types are valid: BR, DL, DR, IL, PT, PX, UB, UP, and UR.

**Other values**

*entry-type*

Specify the entry type that limits the journal entries received. Only journal entries that contain the specified entry type are received. Up to 300 valid entry types can be specified. More information on entry types is in the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

---

**Job name (JOB)**

Specifies the journal entries considered for reception based on their associated jobs.

**Single values**

- **ALL** The journal entries received are not limited to those for a specified job.
- **** The journal entries received are limited to those for the current job.

**Other values**

*job-identifier*

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

*job-name*

Specify the job name of the job.

*user-name*

Specify the user name of the job.

*job-number*

Specify the system-assigned job number.

---

**Program (PGM)**

Specifies the journal entries considered for reception based on their associated programs.

- **ALL** The journal entries received are not limited to those created by a specified program.
**program-name**

Specify the name of the program whose journal entries are considered for reception.

**User profile (USRPRF)**

Specifies that the journal entries considered for reception are limited to the journal entries created for the specified user profile.

*ALL  The journal entries received are not limited to those for a specified user profile.

**user-profile-name**

Specify the name of the user profile whose journal entries are considered for reception.

**Commit cycle large identifier (CCIDLRG)**

Specifies the journal entries considered for reception based on their associated commit cycle identifier. A commit cycle consists of all journal entries sharing the same commit cycle identifier. A journal entry’s commit cycle identifier can be displayed by using the Display Journal (DSPJRN) command and entering option five.

**Note:** You can input a value for either the Commit cycle identifier field (CMTCYCID) or the Commit cycle large identifier field (CCIDLRG) but not for both.

*ALL  The journal entries received are not limited to a specified commit cycle identifier.

**commit-cycle-identifier**

Specify the commit cycle identifier of the journal entries to be considered for reception. The possible range is 1 to 18,446,744,073,709,551,600.

**Dependent entries (DEPENT)**

Specifies whether to receive the journal entries recording actions

• that occur as a result of a trigger program
• on records that are part of a referential constraint
• that will be ignored during an Apply Journaled Changes (APYJRJCHG) or Remove Journaled Changes (RMVJRJCHG) operation.

*ALL  The journal entries relating to trigger programs, referential constraints and the entries which will be ignored by an Apply or Remove Journaled Changes operations are received.

*NONE  The journal entries relating to trigger programs, referential constraints and the entries which will be ignored by an Apply or Remove Journaled Changes operations are not received.
Entry format (ENTFMT)

Specifies the format of the journal entries being received. For a description of what is represented by each of the fields in the journal entry, see the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Note: If ENTFMT(*TYPE1) or ENTFMT(*TYPE2) is not specified, the NULLINDLEN parameter must be specified.

Note: If the Receiver size options field (RCVSIZOPT) for the journal was specified as *MAXOPT3, the sequence number, commit cycle identifier, the count of entries applied or removed or relative record number fields can reach a maximum value of 18,446,744,073,709,551,600. The length of these fields in the ENTFMT(*TYPE1), ENTFMT(*TYPE2), ENTFMT(*TYPE3), and ENTFMT(*TYPE4) formats is defined to hold a 10 digit number. If a sequence number, commit cycle identifier, or count of entries applied or removed or relative record number larger than 10 digits is found and one of these ENTFMT options is specified, the field is set to -1 for that entry.

*TYPE1

The journal entries received are formatted to include the minimum information that can be specified. The information fields and the format of the information in each journal entry is shown below:
### Table 2. Figure: Table 1 - *TYPE1 Journal Entry Format

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Length</td>
<td>5</td>
<td>:1</td>
<td>:5</td>
</tr>
<tr>
<td>Sequence Number (2)</td>
<td>10</td>
<td>:6</td>
<td>:15</td>
</tr>
<tr>
<td>Journal Code</td>
<td>1</td>
<td>:16</td>
<td>:16</td>
</tr>
<tr>
<td>Journal Entry Type</td>
<td>2</td>
<td>:17</td>
<td>:18</td>
</tr>
<tr>
<td>Date</td>
<td>6</td>
<td>:19</td>
<td>:24</td>
</tr>
<tr>
<td>Time</td>
<td>6</td>
<td>:25</td>
<td>:30</td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>:31</td>
<td>:40</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>:41</td>
<td>:50</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>:51</td>
<td>:56</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>:57</td>
<td>:66</td>
</tr>
<tr>
<td>Object Name</td>
<td>10</td>
<td>:67</td>
<td>:76</td>
</tr>
<tr>
<td>Object Library</td>
<td>10</td>
<td>:77</td>
<td>:86</td>
</tr>
<tr>
<td>Member Name</td>
<td>10</td>
<td>:87</td>
<td>:96</td>
</tr>
<tr>
<td>Count/RRN (3)</td>
<td>10</td>
<td>:97</td>
<td>:106</td>
</tr>
<tr>
<td>Flag</td>
<td>1</td>
<td>:107</td>
<td>:107</td>
</tr>
<tr>
<td>Commit Cycle ID (4)</td>
<td>10</td>
<td>:108</td>
<td>:117</td>
</tr>
<tr>
<td>Incomplete data</td>
<td>1</td>
<td>:118</td>
<td>:118</td>
</tr>
<tr>
<td>Minimized Entry</td>
<td>1</td>
<td>:119</td>
<td>:119</td>
</tr>
<tr>
<td>Specific Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>6</td>
<td>:120</td>
<td>:125</td>
</tr>
<tr>
<td>Entry-Specific Data</td>
<td>N (1)</td>
<td>:126</td>
<td>:N+125</td>
</tr>
</tbody>
</table>

**Notes:**
1. The length of the entry-specific data field varies from entry to entry. It is long enough to accommodate all the entry-specific data in each received journal entry.
2. When the RCVSZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
3. When the RCVSZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
4. When the RCVSZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

### *TYPE2

The journal entries received include the information returned when ENTFMT(*TYPE1) is specified, the user profile field, which gives the name of the user who caused the logging of the received journal entries, and the name of the system on which the entry was sent. The format for *TYPE2 journal entries is shown below:
Table 3. Figure: Table 2 - *TYPE2 Journal Entry Format

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Length</td>
<td>5</td>
</tr>
<tr>
<td>Sequence Number (2)</td>
<td>10</td>
</tr>
<tr>
<td>Journal Code</td>
<td>1</td>
</tr>
<tr>
<td>Journal Entry Type</td>
<td>2</td>
</tr>
<tr>
<td>Date</td>
<td>6</td>
</tr>
<tr>
<td>Time</td>
<td>6</td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
</tr>
<tr>
<td>Object Name</td>
<td>10</td>
</tr>
<tr>
<td>Object Library</td>
<td>10</td>
</tr>
<tr>
<td>Member Name</td>
<td>10</td>
</tr>
<tr>
<td>Count/RRN (3)</td>
<td>10</td>
</tr>
<tr>
<td>Flag</td>
<td>1</td>
</tr>
<tr>
<td>Commit Cycle ID (4)</td>
<td>10</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
</tr>
<tr>
<td>System Name</td>
<td>8</td>
</tr>
<tr>
<td>Incomplete data</td>
<td>1</td>
</tr>
<tr>
<td>Minimized Entry</td>
<td>1</td>
</tr>
<tr>
<td>Specific Data</td>
<td>:</td>
</tr>
<tr>
<td>Reserved</td>
<td>18</td>
</tr>
<tr>
<td>Entry-Specific Data</td>
<td>N (1)</td>
</tr>
</tbody>
</table>

Notes:
(1) The length of the entry-specific data field varies from entry to entry. It is long enough to accommodate all the entry-specific data in each received journal entry.
(2) When the RCVSIZOPT of the journal is MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
(3) When the RCVSIZOPT of the journal is MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
(4) When the RCVSIZOPT of the journal is MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

*TYPE3

The journal entries received include the information returned when ENTFMT(*TYPE2) is specified, and the null value indicators. The format of the received entries depends on the value specified on the NULLINDLEN parameter. The tables in the NULLINDLEN parameter description show the three formats for *TYPE3.

*TYPE4

The journal entries received include the information returned when ENTFMT(*TYPE3) is
specified, the journal identifier, the physical file trigger indicator, and the referential constraint indicator. The format of the received entries depends on the value specified on the NULLINDLEN parameter. The tables in the NULLINDLEN parameter description show the three formats for *TYPE4.

*TYPEPTR
The journal entries received include the information as if ENTFMT(*JRNENTFMT) and JRNENTFMT(RJNE0100) were specified, and the entry specific data could contain pointers for specific journal entry types. The format of the received entries depends on the value specified on the NULLINDLEN parameter. The tables in the NULLINDLEN parameter description show the two formats for *TYPEPTR.

Note: NULLINDLEN(*VARLEN) and ENTFMT(*TYPEPTR) cannot be specified at the same time.

*JRNENTFMT
The format of the received journal entries is determined by the JRNENTFMT parameter. The RTN PTR parameter indicates if the received journal entries could contain pointers. The format of the received entries depends on the value specified on the NULLINDLEN parameter. The tables in the NULLINDLEN parameter description show the six formats for *JRNENTFMT.

Format minimized data (FMTMINDTA)
Specifies whether entry specific data which has been minimized on field boundaries will be returned in a readable format.

*NO  The journal entries which have entry specific data that has been minimized on field boundaries will not be returned in a readable format. Therefore, the entry specific data may not be viewable.

*YES  The journal entries which have entry specific data that has been minimized on field boundaries will be returned in a readable format. Therefore, the entry specific data is viewable and may be used for auditing purposes. The fields that were changed are accurately reflected. The fields that were not changed and were not recorded, display default data and are indicated by a value of ‘F9’X in the null value indicators field.

Null value indicators length (NULLINDLEN)
Specifies the length, in bytes, used for the Null Value Indicators portion of the journal entry received by the user. This parameter is not valid if ENTFMT(*TYPE1) or ENTFMT(*TYPE2) is specified.

Null value indicators are present in journal entries for record level operations as follows:
1. The corresponding physical file has null capable fields.
2. The record image has been minimized in the entry specific data.

If the record image has not been minimized in the entry specific data, then there is one null value indicator per field in the physical file. Each indicator is one character long and can be either:
- ‘F0’X = Corresponding field is not null.
- ‘F1’X = Corresponding field is null.

If the record image has been minimized on file field boundaries in the entry specific data and FMTMINDTA(*YES) was specified on the RCVJRNE command, then there is one null value indicator per field in the physical file. Each indicator is one character long and can be either:
- ‘F0’X = Corresponding field is not null.
• 'F1'X = Corresponding field is null.
• 'F9'X = Corresponding field was not changed and the default value for the field is returned.

If the record image has been minimized on file field boundaries in the entry specific data and FMTMINDTA(*NO) was specified on the RCVJRNE command, then an internal value is returned for the null value indicator.

**Single values**

*ENTFMT

The null value indicators field is long enough to include all of the null value indicators in the received journal entries. Since the number of null value indicators can vary from entry to entry, the length of the null value indicators field also varies with each entry.

**Notes:**

If users select the *TYPE3 format, the following information is not available in this format:

1. Incomplete Data indicating if the journal entry data is incomplete due to either LOB fields or Byte Stream File write operations.
2. Minimized Entry Specific Data indicating if the journal entry has minimized entry specific data because the journal had MINENTDTA specified for the object type of the journal entry.

See the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information on the incomplete data indicator, the minimized entry specific data indicator, and these journal entries.

The format for *TYPE3 journal entries when NULLINDLEN(*ENTFMT) is specified is shown below:
Table 4. Table 3 - NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*TYPE3)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>: Length</th>
<th>: From</th>
<th>: To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Length</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Sequence Number (5)</td>
<td>10</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>Journal Code</td>
<td>1</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Journal Entry Type</td>
<td>2</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>Timestamp</td>
<td>26</td>
<td>19</td>
<td>44</td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>45</td>
<td>54</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>55</td>
<td>64</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>65</td>
<td>70</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>71</td>
<td>80</td>
</tr>
<tr>
<td>Object Name</td>
<td>10</td>
<td>81</td>
<td>90</td>
</tr>
<tr>
<td>Object Library</td>
<td>10</td>
<td>91</td>
<td>100</td>
</tr>
<tr>
<td>Member Name</td>
<td>10</td>
<td>101</td>
<td>110</td>
</tr>
<tr>
<td>Count/RNN (6)</td>
<td>10</td>
<td>111</td>
<td>120</td>
</tr>
<tr>
<td>Flag</td>
<td>1</td>
<td>121</td>
<td>121</td>
</tr>
<tr>
<td>Commit Cycle ID (7)</td>
<td>10</td>
<td>122</td>
<td>131</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
<td>132</td>
<td>141</td>
</tr>
<tr>
<td>System Name</td>
<td>8</td>
<td>142</td>
<td>149</td>
</tr>
<tr>
<td>Number of Null Value</td>
<td>5</td>
<td>150</td>
<td>154</td>
</tr>
<tr>
<td>Indicators (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Null Value Indicators</td>
<td>M (2)</td>
<td>155</td>
<td>154 + M</td>
</tr>
<tr>
<td>Length of</td>
<td>5</td>
<td>155 + M</td>
<td>150 + M</td>
</tr>
<tr>
<td>Entry-Specific Data (3):</td>
<td>:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entry-Specific Data</td>
<td>N (4)</td>
<td>160 + M</td>
<td>159 + M</td>
</tr>
</tbody>
</table>

Notes:
1. This field contains the number of null value indicators (in decimal digits) in the received journal entry.
2. The length of null value indicators can vary from entry to entry and is designated by the variable M.
3. This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.
4. The length of entry-specific data can vary from entry to entry and is designated by the variable N.
5. When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
6. When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
7. When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

The format for *TYPE4 journal entries when NULLINDLEN(*ENTFMT) is specified is shown below:

468 IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPHT (Merge TCP/IP Host Table)
Table 5. Figure: Table 4 - NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*TYPE4)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Length</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Sequence Number (5)</td>
<td>10</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>Journal Code</td>
<td>1</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Journal Entry Type</td>
<td>2</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>Timestamp</td>
<td>26</td>
<td>19</td>
<td>44</td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>45</td>
<td>54</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>55</td>
<td>64</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>65</td>
<td>70</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>71</td>
<td>80</td>
</tr>
<tr>
<td>Object Name</td>
<td>10</td>
<td>81</td>
<td>90</td>
</tr>
<tr>
<td>Object Library</td>
<td>10</td>
<td>91</td>
<td>100</td>
</tr>
<tr>
<td>Member Name</td>
<td>10</td>
<td>101</td>
<td>110</td>
</tr>
<tr>
<td>Count/RRN (6)</td>
<td>10</td>
<td>111</td>
<td>120</td>
</tr>
<tr>
<td>Flag</td>
<td>1</td>
<td>121</td>
<td>121</td>
</tr>
<tr>
<td>Commit Cycle ID (7)</td>
<td>10</td>
<td>122</td>
<td>131</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
<td>132</td>
<td>141</td>
</tr>
<tr>
<td>System Name</td>
<td>8</td>
<td>142</td>
<td>149</td>
</tr>
<tr>
<td>Journal Identifier</td>
<td>10</td>
<td>150</td>
<td>159</td>
</tr>
<tr>
<td>Referential Constraint</td>
<td>1</td>
<td>160</td>
<td>160</td>
</tr>
<tr>
<td>Trigger</td>
<td>1</td>
<td>161</td>
<td>161</td>
</tr>
<tr>
<td>Incomplete data</td>
<td>1</td>
<td>162</td>
<td>162</td>
</tr>
<tr>
<td>Ignore during APYJRNCHG or RMVJRNCHG</td>
<td>1</td>
<td>163</td>
<td>163</td>
</tr>
<tr>
<td>Minimized Entry</td>
<td>1</td>
<td>164</td>
<td>164</td>
</tr>
<tr>
<td>Specific Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>5</td>
<td>165</td>
<td>169</td>
</tr>
<tr>
<td>Number of Null Value</td>
<td>5</td>
<td>170</td>
<td>174</td>
</tr>
<tr>
<td>Indicators (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Null Value Indicators</td>
<td>M (2)</td>
<td>175</td>
<td>174 + M</td>
</tr>
<tr>
<td>Length of</td>
<td>5</td>
<td>175 + M</td>
<td>179 + M</td>
</tr>
<tr>
<td>Entry-Specific Data (3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entry-Specific Data</td>
<td>N (4)</td>
<td>180 + M</td>
<td>179 + M</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
1. This field contains the number of null value indicators (in decimal digits) in the received journal entry.
2. The length of null value indicators can vary from entry to entry and is designated by the variable M.
3. This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.
4. The length of entry-specific data can vary from entry to entry and is designated by the variable N.
5. When the RCVSIZOPT of the journal is +MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
6. When the RCVSIZOPT of the journal is +MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
The format for *TYPEPTR or *JRNENTFMT journal entries when NULLINDLEN(*ENTFMT), JRNENTFMT(RJNE0100), and RTNPTR(*SYSMNG) are specified is shown below:
Table 6. Figure: Table 5 - NULLINDLEN(ENTFMT) Journal Entry Format for ENTFMT('TYPEPTR') or ENTFMT('JRNENTFMT') when JRNENTFMT(RJNEO100) is specified

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displacement to next</td>
<td>4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Journal entry's</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>header (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Journal entry's null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>value indicators (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>Journal entry's entry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>specific data (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pointer handle (2)</td>
<td>4</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>Sequence Number</td>
<td>20</td>
<td>16</td>
<td>35</td>
</tr>
<tr>
<td>Journal Code</td>
<td>1</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>Entry Type</td>
<td>2</td>
<td>37</td>
<td>38</td>
</tr>
<tr>
<td>Time stamp</td>
<td>26</td>
<td>39</td>
<td>64</td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>65</td>
<td>74</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>75</td>
<td>84</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>85</td>
<td>90</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>91</td>
<td>100</td>
</tr>
<tr>
<td>Object</td>
<td>30</td>
<td>101</td>
<td>130</td>
</tr>
<tr>
<td>Count/RRN</td>
<td>10</td>
<td>131</td>
<td>140</td>
</tr>
<tr>
<td>Indicator Flag</td>
<td>1</td>
<td>141</td>
<td>141</td>
</tr>
<tr>
<td>Commit Cycle ID</td>
<td>20</td>
<td>142</td>
<td>161</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
<td>162</td>
<td>171</td>
</tr>
<tr>
<td>System Name</td>
<td>8</td>
<td>172</td>
<td>179</td>
</tr>
<tr>
<td>Journal Identifier</td>
<td>10</td>
<td>180</td>
<td>189</td>
</tr>
<tr>
<td>Referential Constraint</td>
<td>1</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>Trigger</td>
<td>1</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>Incomplete Data</td>
<td>1</td>
<td>192</td>
<td>192</td>
</tr>
<tr>
<td>Object Name Indicator</td>
<td>1</td>
<td>193</td>
<td>193</td>
</tr>
<tr>
<td>Ignore During APYJRNCHG:</td>
<td>1</td>
<td>194</td>
<td>194</td>
</tr>
<tr>
<td>or RMJRNCHG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimized Entry Specific Data</td>
<td>1</td>
<td>195</td>
<td>195</td>
</tr>
<tr>
<td>Length Of Null Value</td>
<td>4</td>
<td>NVI Disp</td>
<td>NVI Disp Indicators (1)</td>
</tr>
<tr>
<td>Null Value Indicators</td>
<td>M (4)</td>
<td>NVI Disp</td>
<td>NVI Disp</td>
</tr>
<tr>
<td>Length Of Entry Specific Data</td>
<td>5</td>
<td>ESD Disp</td>
<td>ESD Disp</td>
</tr>
<tr>
<td>Reserved</td>
<td>11</td>
<td>ESD Disp</td>
<td>ESD Disp</td>
</tr>
<tr>
<td>Entry-Specific Data</td>
<td>N (6)</td>
<td>ESD Disp</td>
<td>ESD Disp</td>
</tr>
</tbody>
</table>

Notes:
(1) This field is represented in BINARY(4).
(2) This field is represented in UNSIGNED BINARY(4).
The format for *JRNENTFMT journal entries when NULLINDLEN(*ENTFMT) and JRNENTFMT(RJNE0200) are specified is shown below:
Table 7. Figure: Table 6 - NULLINDLEN('ENTFMT) Journal Entry Format for ENTFMT('JRNENTFMT) when JRNENTFMT(RJNE0200) is specified

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displacement to next</td>
<td>4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Journal entry's</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>header (5)</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Journal entry's null</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>value indicators (5)</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>Journal entry's entry</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>specific data (5)</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>Journal entry's trans-</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>action identifier (1)</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>Journal entry's logical</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>unit of work (I)</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>20</td>
<td>23</td>
</tr>
<tr>
<td>Journal entry's receiver information (I)</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Sequence Number (1)</td>
<td>8</td>
<td>24</td>
<td>31</td>
</tr>
<tr>
<td>Unformatted Time stamp (I)</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Thread Identifier (1)</td>
<td>8</td>
<td>40</td>
<td>47</td>
</tr>
<tr>
<td>System Sequence</td>
<td>8</td>
<td>48</td>
<td>55</td>
</tr>
<tr>
<td>Number (I)</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Count/RRN (1)</td>
<td>8</td>
<td>56</td>
<td>63</td>
</tr>
<tr>
<td>Commit Cycle ID (1)</td>
<td>8</td>
<td>64</td>
<td>71</td>
</tr>
<tr>
<td>Pointer Handle (1)</td>
<td>4</td>
<td>72</td>
<td>75</td>
</tr>
<tr>
<td>Remote Port (1)</td>
<td>2</td>
<td>76</td>
<td>77</td>
</tr>
<tr>
<td>Arm Number (1)</td>
<td>2</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td>Program Library</td>
<td>2</td>
<td>80</td>
<td>81</td>
</tr>
<tr>
<td>ASP Number (1)</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Remote Address</td>
<td>16</td>
<td>82</td>
<td>97</td>
</tr>
<tr>
<td>Journal Code</td>
<td>5</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>Entry Type</td>
<td>99</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>101</td>
<td>110</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>111</td>
<td>120</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>121</td>
<td>126</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>127</td>
<td>136</td>
</tr>
<tr>
<td>Program Library Name</td>
<td>10</td>
<td>137</td>
<td>146</td>
</tr>
<tr>
<td>Program Library ASP</td>
<td>10</td>
<td>147</td>
<td>156</td>
</tr>
<tr>
<td>Device Name</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Object</td>
<td>30</td>
<td>157</td>
<td>186</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
<td>187</td>
<td>196</td>
</tr>
<tr>
<td>Journal Identifier</td>
<td>10</td>
<td>197</td>
<td>206</td>
</tr>
<tr>
<td>Address Family</td>
<td>1</td>
<td>207</td>
<td>207</td>
</tr>
<tr>
<td>System Name</td>
<td>8</td>
<td>208</td>
<td>215</td>
</tr>
<tr>
<td>Indicator Flag</td>
<td>1</td>
<td>216</td>
<td>216</td>
</tr>
</tbody>
</table>

Receive Journal Entry (RCVJRNE) 473
**field-length**

Specify the field length of the null value indicators portion of the received journal entry. Valid values range from 1 through 8000 characters.

The format for *TYPE3 journal entries when NULLINDLEN(field-length) is specified is shown below:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Notes:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1) The length of the null value indicators field is the length specified on the NULLINDLEN parameter.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2) The length of the entry-specific data field varies from entry to entry and is designated by the variable M. This length accommodates all the entry-specific data in each received journal entry.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence Number (3)</td>
<td>10</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>Journal Code</td>
<td>10</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Journal Entry Type</td>
<td>10</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>Timestamp</td>
<td>26</td>
<td>19</td>
<td>44</td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>55</td>
<td>64</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>65</td>
<td>70</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>Object Name</td>
<td>10</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>Object Library</td>
<td>10</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>Member Name</td>
<td>10</td>
<td>91</td>
<td>100</td>
</tr>
<tr>
<td>Count/RRN (4)</td>
<td>10</td>
<td>111</td>
<td>120</td>
</tr>
<tr>
<td>Flag</td>
<td>1</td>
<td>121</td>
<td>121</td>
</tr>
<tr>
<td>Commit Cycle ID (5)</td>
<td>10</td>
<td>122</td>
<td>131</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
<td>132</td>
<td>141</td>
</tr>
<tr>
<td>System Name</td>
<td>8</td>
<td>142</td>
<td>149</td>
</tr>
<tr>
<td>Null Value Indicators</td>
<td>150</td>
<td>149</td>
<td></td>
</tr>
<tr>
<td>Entry-Specific Data</td>
<td>150</td>
<td>149</td>
<td></td>
</tr>
</tbody>
</table>

Table 8. Figure: Table 7 - NULLINDLEN(field-length) Journal Entry Format for ENTFMT(*TYPE3)
The format for *TYPE4 journal entries when NULLINDLEN(field-length) is specified is shown below:
### Table 9. Figure: Table 8 - NULLINDLEN(field-length) Journal Entry Format for ENTFMT(*TYPE4)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Length</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Sequence Number (3)</td>
<td>10</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>Journal Code</td>
<td>1</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Journal Entry Type</td>
<td>2</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>Timestamp</td>
<td>26</td>
<td>19</td>
<td>44</td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>45</td>
<td>54</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>55</td>
<td>64</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>65</td>
<td>70</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>71</td>
<td>80</td>
</tr>
<tr>
<td>Object Name</td>
<td>10</td>
<td>81</td>
<td>90</td>
</tr>
<tr>
<td>Object Library</td>
<td>10</td>
<td>91</td>
<td>100</td>
</tr>
<tr>
<td>Member Name</td>
<td>10</td>
<td>101</td>
<td>110</td>
</tr>
<tr>
<td>Count/RNN (4)</td>
<td>10</td>
<td>111</td>
<td>120</td>
</tr>
<tr>
<td>Flag</td>
<td>1</td>
<td>121</td>
<td>121</td>
</tr>
<tr>
<td>Commit Cycle ID (5)</td>
<td>10</td>
<td>122</td>
<td>131</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
<td>132</td>
<td>141</td>
</tr>
<tr>
<td>System Name</td>
<td>8</td>
<td>142</td>
<td>149</td>
</tr>
<tr>
<td>Journal Identifier</td>
<td>10</td>
<td>150</td>
<td>159</td>
</tr>
<tr>
<td>Referential Constraint</td>
<td>1</td>
<td>160</td>
<td>160</td>
</tr>
<tr>
<td>Trigger</td>
<td>1</td>
<td>161</td>
<td>161</td>
</tr>
<tr>
<td>Incomplete data</td>
<td>1</td>
<td>162</td>
<td>162</td>
</tr>
<tr>
<td>Ignore during APYRNCHG or RMVJRNCCHG</td>
<td>1</td>
<td>163</td>
<td>163</td>
</tr>
<tr>
<td>Minimized Entry</td>
<td>1</td>
<td>164</td>
<td>164</td>
</tr>
<tr>
<td>Specific Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>5</td>
<td>165</td>
<td>169</td>
</tr>
<tr>
<td>Null Value Indicators</td>
<td>field</td>
<td>170</td>
<td>169</td>
</tr>
<tr>
<td></td>
<td>length</td>
<td>169</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entry-Specific Data</td>
<td>M (2)</td>
<td>170</td>
<td>M+</td>
</tr>
<tr>
<td></td>
<td>Field</td>
<td>169</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>length</td>
<td>169</td>
<td></td>
</tr>
</tbody>
</table>

Notes:
(1) The length of the null value indicators field is the length specified on the NULLINDLEN parameter.
(2) The length of the entry-specific data field varies from entry to entry and is designated by the variable M. This length accommodates all the entry-specific data in each received journal entry.
(3) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
(4) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
(5) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the entry-specific data is larger than 10 digits.
The format for *TYPEPTR or *JRNENTFMT journal entries when NULLINDLEN(field-length) and JRNENTFMT(RJNE0100) are specified is shown below:
Table 10. Figure: Table 9 - NULLINDLEN(field-length) Journal Entry Format for ENTFMT(*TYPEPTR) or ENTFMT(*JRNENTFMT) when JRNENTFMT(RJNE0100) is specified

<table>
<thead>
<tr>
<th>Field Name</th>
<th>: Length : From : To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displacement to next</td>
<td>: 4 : 0 : 3</td>
</tr>
<tr>
<td>journal entry's header (1)</td>
<td>: : :</td>
</tr>
<tr>
<td>Displacement to this</td>
<td>: 4 : 4 : 7</td>
</tr>
<tr>
<td>journal entry's null value indicators (1)</td>
<td>: : :</td>
</tr>
<tr>
<td>Displacement to this</td>
<td>: 4 : 8 : 11</td>
</tr>
<tr>
<td>journal entry's entry specific data (1)</td>
<td>: : :</td>
</tr>
<tr>
<td>Pointer handle (2)</td>
<td>: 4 : 12 : 15</td>
</tr>
<tr>
<td>Sequence Number</td>
<td>: 20 : 16 : 35</td>
</tr>
<tr>
<td>Journal Code</td>
<td>: 1 : 36 : 36</td>
</tr>
<tr>
<td>Entry Type</td>
<td>: 2 : 37 : 38</td>
</tr>
<tr>
<td>Time stamp</td>
<td>: 26 : 39 : 64</td>
</tr>
<tr>
<td>Job Name</td>
<td>: 10 : 65 : 74</td>
</tr>
<tr>
<td>User Name</td>
<td>: 10 : 75 : 84</td>
</tr>
<tr>
<td>Job Number</td>
<td>: 6 : 85 : 90</td>
</tr>
<tr>
<td>Program Name</td>
<td>: 10 : 91 : 100</td>
</tr>
<tr>
<td>Object</td>
<td>: 30 : 101 : 130</td>
</tr>
<tr>
<td>Count/RN</td>
<td>: 10 : 131 : 140</td>
</tr>
<tr>
<td>Indicator Flag</td>
<td>: 1 : 141 : 141</td>
</tr>
<tr>
<td>Commit Cycle ID</td>
<td>: 20 : 142 : 161</td>
</tr>
<tr>
<td>User Profile</td>
<td>: 10 : 162 : 171</td>
</tr>
<tr>
<td>System Name</td>
<td>: 0 : 172 : 179</td>
</tr>
<tr>
<td>Journal Identifier</td>
<td>: 10 : 180 : 189</td>
</tr>
<tr>
<td>Referential Constraint</td>
<td>: 1 : 190 : 190</td>
</tr>
<tr>
<td>Trigger</td>
<td>: 1 : 191 : 191</td>
</tr>
<tr>
<td>Incomplete Data</td>
<td>: 1 : 192 : 192</td>
</tr>
<tr>
<td>Object Name Indicator</td>
<td>: 1 : 193 : 193</td>
</tr>
<tr>
<td>Ignore During APYJRNCHG</td>
<td>: 1 : 194 : 194</td>
</tr>
<tr>
<td>or RMJRNCHG</td>
<td>: : :</td>
</tr>
<tr>
<td>Minimized Entry Specific Data</td>
<td>: 1 : 195 : 195</td>
</tr>
<tr>
<td>Null Value Indicators</td>
<td>: length : NVI Disp : NVI Disp</td>
</tr>
<tr>
<td>: length : (4) : + field</td>
<td></td>
</tr>
<tr>
<td>: (3) : : length</td>
<td></td>
</tr>
<tr>
<td>: : :</td>
<td></td>
</tr>
<tr>
<td>Length Of Entry</td>
<td>: 5 : ESD Disp : ESD Disp</td>
</tr>
<tr>
<td>Specific Data</td>
<td>: (5) : + 4</td>
</tr>
<tr>
<td>Reserved</td>
<td>: 11 : ESD Disp : ESD Disp</td>
</tr>
<tr>
<td>: : 5 : + 15</td>
<td></td>
</tr>
<tr>
<td>Entry-Specific Data</td>
<td>: N (6) : ESD Disp : ESD Disp</td>
</tr>
<tr>
<td>: : 16 : + 15 + N</td>
<td></td>
</tr>
</tbody>
</table>

478es: IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPIHT (Merge TCP/IP Host Table)
(1) This field is represented in BINARY(4).
(2) This field is represented in UNSIGNED BINARY(4).
(3) The length of the null value indicators field is the length specified on the NULLINDLEN parameter.
The format for *JRNENTFMT journal entries when NULLINDLEN(field-length) and JRNENTFMT(RJNE0200) are specified is shown below:
Table 11. Figure: Table 10 - NULLINDLEN(field-length) Journal Entry Format for ENTFMT(*JRNENTFMT) when JRNENTFMT(RJNE0200) is specified

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displacement to next</td>
<td>4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Journal entry's header (5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Journal entry's null value indicators (5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>Journal entry's entry specific data (5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>Journal entry's transaction identifier (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>Journal entry's logical unit of work (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>20</td>
<td>23</td>
</tr>
<tr>
<td>Journal entry's receiver information (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence Number (1)</td>
<td>8</td>
<td>24</td>
<td>31</td>
</tr>
<tr>
<td>Unformatted Time stamp (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thread Identifier (1)</td>
<td>8</td>
<td>40</td>
<td>47</td>
</tr>
<tr>
<td>System Sequence Number (1)</td>
<td>8</td>
<td>48</td>
<td>55</td>
</tr>
<tr>
<td>Count/RNN (1)</td>
<td>8</td>
<td>56</td>
<td>63</td>
</tr>
<tr>
<td>Commit Cycle ID (1)</td>
<td>8</td>
<td>64</td>
<td>71</td>
</tr>
<tr>
<td>Pointer Handle (1)</td>
<td>4</td>
<td>72</td>
<td>75</td>
</tr>
<tr>
<td>Remote Port (1)</td>
<td>2</td>
<td>76</td>
<td>77</td>
</tr>
<tr>
<td>Arm Number (1)</td>
<td>2</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td>Program Library Number (1)</td>
<td>2</td>
<td>80</td>
<td>81</td>
</tr>
<tr>
<td>ASP Number (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote Address</td>
<td>16</td>
<td>82</td>
<td>97</td>
</tr>
<tr>
<td>Journal Code</td>
<td>1</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>Entry Type</td>
<td>2</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>101</td>
<td>110</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>111</td>
<td>120</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>121</td>
<td>126</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>127</td>
<td>136</td>
</tr>
<tr>
<td>Program Library Name</td>
<td>10</td>
<td>137</td>
<td>146</td>
</tr>
<tr>
<td>Program Library ASP</td>
<td>10</td>
<td>147</td>
<td>156</td>
</tr>
<tr>
<td>Device Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object</td>
<td>30</td>
<td>157</td>
<td>186</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
<td>187</td>
<td>196</td>
</tr>
<tr>
<td>Journal Identifier</td>
<td>10</td>
<td>197</td>
<td>206</td>
</tr>
</tbody>
</table>

Starting with MRGTCPHT (Merge TCP/IP Host Table)
Element 1: Field data format

**VARLEN**

The null value indicators field is a variable length field.

*field-length*

Specify the field length of the null value indicators portion of the received journal entry. Valid values range from 1 through 8000 characters.

Element 2: Variable length field length

*maximum-field-length*

Specify the maximum field length of the null value indicators portion of the received journal entry. Valid values range from 1 to 8000 characters. If a journal entry has more null value indicators than the value you specify and truncation results in the loss of a non 'F0'X indicator value, the RCVJRNE request is abnormally ended.

You can specify this element only if you also specify *VARLEN on the first element of this parameter.

Note: NULLINDLEN(*VARLEN) and ENTFMT(*TYPEPTR) cannot be specified at the same time.

Note: NULLINDLEN(*VARLEN) and RTNPTR(*SYSMNG) or RTNPTR(*USRMNG) cannot be specified at the same time.

The format for *TYPE3 journal entries when NULLINDLEN(*VARLEN field-length) is specified is shown below:
Table 12: Table 11 - NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTFMT(*TYPE3)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Length</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Sequence Number (4)</td>
<td>10</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>Journal Code</td>
<td>1</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Journal Entry Type</td>
<td>2</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>Timestamp</td>
<td>26</td>
<td>19</td>
<td>44</td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>45</td>
<td>54</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>55</td>
<td>64</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>65</td>
<td>70</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>71</td>
<td>80</td>
</tr>
<tr>
<td>Object Name</td>
<td>10</td>
<td>81</td>
<td>90</td>
</tr>
<tr>
<td>Object Library</td>
<td>10</td>
<td>91</td>
<td>100</td>
</tr>
<tr>
<td>Member Name</td>
<td>10</td>
<td>101</td>
<td>110</td>
</tr>
<tr>
<td>Count/RRN (5)</td>
<td>10</td>
<td>111</td>
<td>120</td>
</tr>
<tr>
<td>Flag (1)</td>
<td>1</td>
<td>121</td>
<td>121</td>
</tr>
<tr>
<td>Commit Cycle ID (6)</td>
<td>10</td>
<td>122</td>
<td>131</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
<td>132</td>
<td>141</td>
</tr>
<tr>
<td>System Name</td>
<td>8</td>
<td>142</td>
<td>149</td>
</tr>
<tr>
<td>Number of Null Value (2)</td>
<td>2</td>
<td>150</td>
<td>151</td>
</tr>
<tr>
<td>Indicators (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Null Value Indicators</td>
<td>152</td>
<td>151</td>
<td>+</td>
</tr>
<tr>
<td>Length of Entry-Specific Data (2):</td>
<td>5</td>
<td>152</td>
<td>156</td>
</tr>
<tr>
<td>Entry-Specific Data (3):</td>
<td>M (3)</td>
<td>157</td>
<td>156 + M</td>
</tr>
</tbody>
</table>

Notes:
(1) This field contains the number of null value indicators (in binary digits) in the received journal entry.
(2) This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.
(3) The length of entry-specific data can vary from entry to entry and is designated by the variable M.
(4) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
(5) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
(6) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

The format for *TYPE4 journal entries when NULLINDLEN(*VARLEN field-length) is specified is shown below:
Table 13. Figure: Table 12 - NULLINDLEN(‘VARLEN field-length) Journal Entry Format for ENTFMT(‘TYPE4)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Length</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Sequence Number (4)</td>
<td>10</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>Journal Code</td>
<td>1</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Journal Entry Type</td>
<td>2</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>Timestamp</td>
<td>26</td>
<td>19</td>
<td>44</td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>45</td>
<td>54</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>55</td>
<td>64</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>65</td>
<td>70</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>71</td>
<td>80</td>
</tr>
<tr>
<td>Object Name</td>
<td>10</td>
<td>81</td>
<td>90</td>
</tr>
<tr>
<td>Object Library</td>
<td>10</td>
<td>91</td>
<td>100</td>
</tr>
<tr>
<td>Member Name</td>
<td>10</td>
<td>101</td>
<td>110</td>
</tr>
<tr>
<td>Count/RRN (5)</td>
<td>10</td>
<td>111</td>
<td>120</td>
</tr>
<tr>
<td>Flag</td>
<td>1</td>
<td>121</td>
<td>121</td>
</tr>
<tr>
<td>Commit Cycle ID (6)</td>
<td>10</td>
<td>122</td>
<td>131</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
<td>132</td>
<td>141</td>
</tr>
<tr>
<td>System Name</td>
<td>8</td>
<td>142</td>
<td>149</td>
</tr>
<tr>
<td>Journal Identifier</td>
<td>10</td>
<td>150</td>
<td>159</td>
</tr>
<tr>
<td>Referential Constraint</td>
<td>1</td>
<td>160</td>
<td>160</td>
</tr>
<tr>
<td>Trigger</td>
<td>1</td>
<td>161</td>
<td>161</td>
</tr>
<tr>
<td>Incomplete data</td>
<td>1</td>
<td>162</td>
<td>162</td>
</tr>
<tr>
<td>Ignore during APVJRNCNG or RMVJRNCNG</td>
<td>1</td>
<td>163</td>
<td>163</td>
</tr>
<tr>
<td>Minimized Entry Specific Data</td>
<td>1</td>
<td>164</td>
<td>164</td>
</tr>
<tr>
<td>Reserved</td>
<td>5</td>
<td>165</td>
<td>169</td>
</tr>
<tr>
<td>Number of Null Value Indicators (1)</td>
<td>2</td>
<td>170</td>
<td>171</td>
</tr>
<tr>
<td>Null Value Indicators</td>
<td>field</td>
<td>172</td>
<td>171 +</td>
</tr>
<tr>
<td></td>
<td>length</td>
<td></td>
<td>field</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>length</td>
</tr>
<tr>
<td>Length of Entry-Specific Data (2):</td>
<td>field</td>
<td>field</td>
<td></td>
</tr>
<tr>
<td></td>
<td>length</td>
<td></td>
<td>length</td>
</tr>
<tr>
<td>Entry-Specific Data (3)</td>
<td>M</td>
<td>177</td>
<td>176 +</td>
</tr>
<tr>
<td></td>
<td>length</td>
<td></td>
<td>length</td>
</tr>
</tbody>
</table>

Notes:
(1) This field contains the number of null value indicators (in binary digits) in the received journal entry.
(2) This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.
(3) The length of entry-specific data can vary from entry to entry and is designated by the variable M.
(4) When the RCVSIZOPT of the journal is +MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
(5) When the RCVSIZOPT of the journal is +MAXOPT3, this field...
The format for *JRNENTFMT journal entries when NULLINDLEN(*VARLEN field-length) and JRNENTFMT(RJNE0100) are specified is shown below:
Table 14. Figure: Table 13 - NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTFMT(*JRNENTFMT) when JRNENTFMT(RJNE0100) is specified

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displacement to next</td>
<td>4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Journal entry's</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>header (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Journal entry's null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>value indicators (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>Journal entry's entry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>specific data (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pointer handle (2)</td>
<td>4</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>Sequence Number</td>
<td>20</td>
<td>16</td>
<td>35</td>
</tr>
<tr>
<td>Journal Code</td>
<td>1</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>Entry Type</td>
<td>2</td>
<td>37</td>
<td>38</td>
</tr>
<tr>
<td>Time stamp</td>
<td>26</td>
<td>39</td>
<td>64</td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>65</td>
<td>74</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>75</td>
<td>84</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>85</td>
<td>90</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>91</td>
<td>100</td>
</tr>
<tr>
<td>Object</td>
<td>30</td>
<td>101</td>
<td>130</td>
</tr>
<tr>
<td>Count/RRN</td>
<td>10</td>
<td>131</td>
<td>140</td>
</tr>
<tr>
<td>Indicator Flag</td>
<td>1</td>
<td>141</td>
<td>141</td>
</tr>
<tr>
<td>Commit Cycle ID</td>
<td>20</td>
<td>142</td>
<td>161</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
<td>162</td>
<td>171</td>
</tr>
<tr>
<td>System Name</td>
<td>8</td>
<td>172</td>
<td>179</td>
</tr>
<tr>
<td>Journal Identifier</td>
<td>10</td>
<td>180</td>
<td>189</td>
</tr>
<tr>
<td>Referential Constraint</td>
<td>1</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>Trigger</td>
<td>1</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>Incomplete Data</td>
<td>1</td>
<td>192</td>
<td>192</td>
</tr>
<tr>
<td>Object Name Indicator</td>
<td>1</td>
<td>193</td>
<td>193</td>
</tr>
<tr>
<td>Ignore During APYJRNCNG:</td>
<td>1</td>
<td>194</td>
<td>194</td>
</tr>
<tr>
<td>or RMVJRNCNG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimized Entry</td>
<td>1</td>
<td>195</td>
<td>195</td>
</tr>
<tr>
<td>Specific Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length Of Null Value</td>
<td>4</td>
<td>NVI Disp</td>
<td>NVI Disp</td>
</tr>
<tr>
<td>Indicators (1)</td>
<td></td>
<td>(3)</td>
<td>+ 3</td>
</tr>
<tr>
<td>Null Value Indicators</td>
<td>field</td>
<td>NVI Disp</td>
<td>NVI Disp</td>
</tr>
<tr>
<td>length</td>
<td>+ 4</td>
<td>+ 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+ field</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length Of Entry</td>
<td>5</td>
<td>ESD Disp</td>
<td>ESD Disp</td>
</tr>
<tr>
<td>Specific Data</td>
<td></td>
<td>(4)</td>
<td>+ 4</td>
</tr>
<tr>
<td>Reserved</td>
<td>11</td>
<td>ESD Disp</td>
<td>ESD Disp</td>
</tr>
<tr>
<td></td>
<td>+ 5</td>
<td>+ 15</td>
<td></td>
</tr>
<tr>
<td>Entry-Specific Data</td>
<td>M (5)</td>
<td>ESD Disp</td>
<td>ESD Disp</td>
</tr>
<tr>
<td></td>
<td>+ 16</td>
<td>+ 15 + M</td>
<td></td>
</tr>
</tbody>
</table>

Notes:
(1) This field is represented in BINARY(4).
The format for *JRNENTFMT journal entries when NULLINDLEN(*VARLEN field-length) and JRNENTFMT(RJNE0200) are specified is shown below:
Table 15. Figure: Table 14 - NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTFMT(*JRNENTFMT) when JRNENTFMT(RJNE0200) is specified

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displacement to next</td>
<td>4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Journal entry's header (5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Journal entry's null value indicators (5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>Journal entry's entry specific data (5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>Journal entry's transaction identifier (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>Journal entry's logical unit of work (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displacement to this</td>
<td>4</td>
<td>20</td>
<td>23</td>
</tr>
<tr>
<td>Journal entry's receiver information (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence Number (1)</td>
<td>8</td>
<td>24</td>
<td>31</td>
</tr>
<tr>
<td>Unformatted Time stamp (1)</td>
<td>8</td>
<td>32</td>
<td>39</td>
</tr>
<tr>
<td>Thread Identifier (1)</td>
<td>8</td>
<td>40</td>
<td>47</td>
</tr>
<tr>
<td>System Sequence Number (1)</td>
<td>8</td>
<td>48</td>
<td>55</td>
</tr>
<tr>
<td>Count/RRN (1)</td>
<td>8</td>
<td>56</td>
<td>63</td>
</tr>
<tr>
<td>Commit Cycle ID (1)</td>
<td>8</td>
<td>64</td>
<td>71</td>
</tr>
<tr>
<td>Pointer Handle (1)</td>
<td>4</td>
<td>72</td>
<td>75</td>
</tr>
<tr>
<td>Remote Port (1)</td>
<td>2</td>
<td>76</td>
<td>77</td>
</tr>
<tr>
<td>Arm Number (1)</td>
<td>2</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td>Program Library ASP Number (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote Address</td>
<td>16</td>
<td>82</td>
<td>97</td>
</tr>
<tr>
<td>Journal Code</td>
<td>1</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>Entry Type</td>
<td>99</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Job Name</td>
<td>10</td>
<td>101</td>
<td>110</td>
</tr>
<tr>
<td>User Name</td>
<td>10</td>
<td>111</td>
<td>120</td>
</tr>
<tr>
<td>Job Number</td>
<td>6</td>
<td>121</td>
<td>126</td>
</tr>
<tr>
<td>Program Name</td>
<td>10</td>
<td>127</td>
<td>136</td>
</tr>
<tr>
<td>Program Library Name</td>
<td>10</td>
<td>137</td>
<td>146</td>
</tr>
<tr>
<td>Program Library ASP</td>
<td>10</td>
<td>147</td>
<td>156</td>
</tr>
<tr>
<td>Device Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object</td>
<td>30</td>
<td>157</td>
<td>186</td>
</tr>
<tr>
<td>User Profile</td>
<td>10</td>
<td>187</td>
<td>196</td>
</tr>
<tr>
<td>Journal Identifier</td>
<td>10</td>
<td>197</td>
<td>206</td>
</tr>
<tr>
<td>Address Family</td>
<td>1</td>
<td>207</td>
<td>207</td>
</tr>
<tr>
<td>System Name</td>
<td>208</td>
<td>215</td>
<td></td>
</tr>
<tr>
<td>Indicator Flag</td>
<td>1</td>
<td>216</td>
<td>216</td>
</tr>
</tbody>
</table>
Delay time (DELAY)

Specifies the number of seconds that the command processing program (CPP) waits for a new journal entry to arrive if the last entry has already been received. After the last entry in the journal is received and passed to the exit program, the CPP tries to receive the next entry. If no new journal entry exists, the exit program is passed a value of 0 in the first byte of the second parameter.

**Note:** This parameter is valid only when TOENTLRG(*NONE) and TOENT(*NONE) is specified, and the last receiver specified on the RCVRNG parameter identifies the journal receiver that is currently attached when journal entries are starting to be received.

When the last entry on the journal has been passed to the exit program and no journal entries are currently available to be passed to the exit program, one of the following occurs:

- If a number of seconds is specified for the first element in the list, the exit program is immediately called and a '0' is passed to the first byte of the second exit parameter indicating that no additional journal entries are currently available. When the exit program returns control to the command, the system delays for the specified number of seconds.

  When the delay time has expired, the system then checks whether any additional journal entries are available to be passed to the exit program. Any additional entries are passed to the exit program sequentially, until there are no more available. When there are no further journal entries available, the exit program is called, and a '0' is passed as the first byte of the second exit program parameter, indicating there are no more journal entries currently available. When the exit program returns control to the command, the system again delays for the specified number of seconds.

- If there are no new journal entries to pass to the exit program after the delay, the exit program is called, and a '0' is passed as the first byte of the second exit program parameter to indicate that no further journal entries are available. The exit program then passes the value '9' for the first byte of the second parameter, indicating that this command is to end.

- If *NEXTENT is specified for the first element in the list, then additional journal entries are passed to the exit program as they become available. When this option is used, the second element in the list indicates the maximum number of seconds between calls to the exit program. If there are no additional journal entries to pass after the specified maximum delay time, the exit program is called, and a '0' is passed to the first byte of the second exit program parameter, indicating that no additional journal entries are currently available.

  The maximum delay time can be either of the following:

  - The time between a call to the exit program passing the last currently available journal entry, and a subsequent call to the exit program indicating that no new journal entries are available.
  - The time between calls to the exit program indicating that no additional journal entries are available.

  If the exit program is called after the maximum delay has expired, it then can pass the value '9' for the first byte of the second parameter, indicating that this command should be ended.

**Note:** The previous description of the DELAY parameter assumes that the journal receiver that is currently attached at the beginning of the RCVJRNE command is still attached. If that journal receiver has been detached, the exit program is sent the reason code 3 after all journal entries have been received by the exit program and the RCVJRNE command ends.

**Element 1: Delay time value**

30 The command delays 30 seconds before checking whether additional journal entries are available to be passed to the exit program.
*NEXTENT
A fixed delay time is not used. Additional journal entries are passed to the exit program as they become available.

Note: If the RCVJRNE exit program causes any additional calls of the RCVJRNE command, those additional calls cannot specify DELAY(*NEXTENT) if a preceding call specified TOENTLRG(*NONE) or TOENT(*NONE).

Note: INCENT(*ALL) and DELAY(*NEXTENT) cannot be specified at the same time.

seconds
Specify the number of seconds that the command delays before checking whether additional journal entries are available to be passed to the exit program. Valid values range from 1 through 99999.

Element 2: Maximum delay time value
This element indicates the maximum number of seconds between calls to the exit program when a fixed delay time is not specified on the first element. This element is valid only if *NEXTENT is specified for the first element.

*CLS  The process default wait time is used as the maximum number of seconds between calls to the exit program.

seconds
Specify the maximum length of time between calls to the exit program, in seconds. Valid values range from 1 through 99999.

Block length (BLKLEN)
Specifies whether the system will be sending one or more journal entries to the exit program and specifies the block length of the buffer passed to the exit program. The EXITPGM parameter has further details about block mode semantics.

*NONE  At most one journal entry will be sent to the exit program.

*CALC  One or more journal entries will be passed to the exit program in a block. The length of the block passed (the first parameter passed to the exit program) is determined by the system and will be optimal.

block-length
Specify the length in kilobytes of the buffer passed to the exit program (EXITPGM parameter). Valid values range from 32 to 4000.

Journal entry format (JRNENTMFT)
Specifies the format of the journal entries received by the exit program. The formats are described in the Retrieve Journal Entries (QjoRetrieveJournalEntries) API. See the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for the details associated with these formats.

This parameter is only valid if ENTFMT(*JRNENTFMT) is also specified.
RJNE0100
Received journal entries are in RJNE0100 format.

RJNE0200
Received journal entries are in RJNE0200 format. While in block mode, a single block of entries will not be received from multiple receivers.

Return pointers (RTNPTR)
Specifies whether the journal entries received include entry specific data that could contain pointers for specific journal entry types.

This parameter is only valid if ENTFMT(*JRNENTFMT) is also specified.

*NONE
Received journal entries will not include pointers.

*SYSMNG
Specifies that journal entries received include entry specific data that could contain pointers for specific journal entry types. The system will manage the releasing of the resources related to pointers that are received in the journal entries. The pointers, their associated pointer handles and any related storage will be released when the exit program returns control to the system.

Note: NULLINDLEN(*VARLEN) and RTNPTR(*SYSMNG) or RTNPTR(*USRMNG) cannot be specified at the same time.

*USRMNG
Specifies that journal entries received include entry specific data that could contain pointers for specific journal entry types. The user will manage the releasing of the resources related to pointers that are received in the journal entries. Neither the pointers, their associated pointer handles or any related storage will be released when the exit program returns control to the system. The user must use the Delete Pointer Handle (QIODeletePointerHandle) API to release these resources. See the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for the details associated with this API.

Note: NULLINDLEN(*VARLEN) and RTNPTR(*SYSMNG) or RTNPTR(*USRMNG) cannot be specified at the same time.

Include entries (INCENT)
Specifies whether only the confirmed or both the confirmed and unconfirmed, journal entries are received. This parameter only applies when receiving journal entries for output from a remote journal.

Confirmed entries are those journal entries which have been sent to this remote journal and the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is known.

Unconfirmed entries are those journal entries which have been sent to this remote journal, but the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is not known, or the object name information for those journal entries is not yet known to the remote journal. Unconfirmed journal entries can only exist within the attached receiver of a remote journal. This only applies if synchronous delivery mode is being used for a particular remote journal.

*CONFIRMED
Only those journal entries which have been confirmed are received.
ALL All confirmed and unconfirmed journal entries are received.

Note: INCENT(*ALL) and DELAY(*NEXTENT) cannot be specified at the same time.

Starting sequence number (FROMENT)
Specifies the first journal entry considered for reception.

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

*FIRST
The first journal entry in the specified journal receiver range is the first entry considered for reception.

starting-sequence-number
Specify the sequence number of the first journal entry considered for reception. The possible range is 1 to 9,999,999,999.

Ending sequence number (TOENT)
Specifies the last journal entry considered for reception.

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.

*NONE
No journal entry is specified. Journal entries are passed to the exit program until the command is canceled (by a cancel request or a cancel job command) or until an end reason code (9) is set by the exit program. If there are no more entries to pass, the RCVJRNE command waits the number of seconds indicated on the DELAY parameter before trying to find more entries to pass.

Note: TOENT(*NONE) is valid only if the RCVRNG parameter specifies a receiver that is currently attached when starting to receive journal entries.

*LAST
The last journal entry in the journal receiver range specified is the last entry considered for reception.

ending-sequence-number
Specify the sequence number of the final journal entry considered for reception. The possible range is 1 to 9,999,999,999.

Note: The values specified for the from and to prompts can be the same. For example, FROMENT(234) and TOENT(234) can be specified.

Commit cycle identifier (CMTCYCID)
Specifies the journal entries considered for reception based on their associated commit cycle identifier. A commit cycle consists of all journal entries sharing the same commit cycle identifier. A journal entry’s commit cycle identifier can be displayed by using the Display Journal (DSPJRN) command and entering option five.
**Note:** You can input a value for either the **Commit cycle identifier** field (CMTCYCID) or the **Commit cycle large identifier** field (CCIDLRG) but not for both.

*ALL*  The journal entries received are not limited to a specified commit cycle identifier.

**commit-cycle-identifier**  
Specify the commit cycle identifier of the journal entries to be considered for reception. The possible range is 1 to 9,999,999,999.

---

### Examples

**Example 1: Receiving Journal Entries**

RCVJRNE  JRN(APPLIB/JRN1)  EXITPGM(MYLIB/RCVPGM)  
  FILE(APPLIB/FILE3)  TOENT(+LAST)  ENTFMT(*TYPE3)  
  NULLINDLEN(*ENTFMT)

This command receives journal entries from the journal receiver that is currently attached (when journal entries are starting to be received) to the journal JRN1 in library APPLIB and passes them one at a time to program RCVPGM in library MYLIB. Only entries with file-level information for the first member of file FILE3 in library APPLIB are received. The format of each entry passed to the exit program is shown in the "NULLINDLEN(*ENTFMT)" Journal Entry Format for ENTFMT(*TYPE3)" table shown within the NULLINDLEN parameter description.

**Example 2: Receiving Journal Entries**

RCVJRNE  JRN(JRNLIB/MYJRN)  EXITPGM(RCVLIB/PGMA)  
  FILE(FILELIB/PFILE MBRONE)  
  TOENT(+LAST)  ENTFMT(*TYPE3)  NULLINDLEN(*VARLEN 30)

This command receives journal entries with file-level information for member MBRONE of file PFILE in library FILELIB from the journal receiver currently attached (when journal entries are starting to be received) to journal MYJRN in library JRNLIB and sends them one at a time to program PGMA in library RCVLIB. The format of each entry passed to the exit program is shown in the “NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTFMT(*TYPE3)” table at the end of the NULLINDLEN parameter description. The null value indicators portion of each received entry is 30 characters in length.

**Example 3: Receiving Journal Entries Using DELAY(*NEXTENT)**

RCVJRNE  JRN(JRNLIB/MYJRN)  EXITPGM(RCVLIB/PGMA)  
  RCVRNG(+CURCHAIN)  TOENT(+NONE)  DELAY(*NEXTENT)

This command receives all available journal entries from the chain of journal receivers, which includes the journal receiver that is attached at the start of receiving journal entries, associated with the journal MYJRN in the library JRNLIB. These journal entries are sent sequentially to exit program PGMA in library RCVLIB, as they become available. The maximum length of time between calls to the exit program is equal to the process default wait time value.

**Example 4: Receiving Journal Entries for Data Area, Data Queue, and Integrated File System Objects**

RCVJRNE  JRN(LIBPROD/PRODJRN)  EXITPGM(RCVLIB/PGMA)  
  OBJ((APPLIB/D1 *DTAARA)  (APPLIB/D2 *DTAQ)  
  (APPLIB/D3 *FILE *NONE))  
  OBJPATH(`/mydirectory`)  SUBTREE(*YES)  
  TOENT(+LAST)  ENTFMT(*JRNENTFMT)  JRNENTFMT(RJNE0200)

This command receives journal entries from the journal receiver currently attached (when journal entries are starting to be received) to journal PRODJRN in library LIBPROD and sends them one at a time to program PGMA in library RCVLIB. The format of each entry passed to the exit program is shown in the
"NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*JRNENTFMT) when
JRNENTFMT(RJNE0200) is specified" table at the end of the NULLINDLEN parameter description. Only
entries associated with the specified objects are converted. These objects are a data area in library APPLIB
called D1, a data queue in library APPLIB called D2, and a database file D3 in library APPLIB, as well as
the directory `/mydirectory` and all directories, stream files, and symbolic links within that directory or
one of its subdirectories.

## Error messages

*ESCAPE Messages

CPF7002
File &1 in library &2 not a physical file.

CPF7006
Member &3 not found in file &1 in &2.

CPF7007
Cannot allocate member &3 file &1 in &2.

CPF701B
Journal recovery of an interrupted operation failed.

CPF705C
INCENT(*ALL) not allowed for a local journal.

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

CPF7054
FROM and TO values not valid.

CPF7055
Maximum number of objects exceeded.

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

CPF706A
Significant null value indicator truncated.

CPF706D
RCVJNE exit program &1 in &2 failed.

CPF7060
Object not found and not journaled in specified receiver range.

CPF7061
Conversion of journal entries failed.

CPF7062
No entries converted or received from journal &1.

CPF7065
Entry type (ENTTYP) not valid for journal code (JRNCDE).

CPF707B
DELAY(*NEXTENT) not allowed.

CPF7074
RCVRNG for specified SEARCH not valid.
CPF708C
  DELAY(*NEXTENT) not allowed with INCENT(*ALL).

CPF708D
  Journal receiver found logically damaged.

CPF709C
  JOB, PGM, and USRPRF not valid for receiver range.

CPF7096
  Ending receiver for RCVRNG must identify an attached receiver.

CPF70A9
  OBJPATH parameter not valid for a remote journal.

CPF70AC
  File identifier &1 not found.

CPF9801
  Object &2 in library &3 not found.

CPF9802
  Not authorized to object &2 in &3.

CPF9803
  Cannot allocate object &2 in library &3.

CPF9809
  Library &1 cannot be accessed.

CPF9810
  Library &1 not found.

CPF9820
  Not authorized to use library &1.

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

CPF9825
  Not authorized to device &1.
Receive Message (RCVMSG)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Yes

The Receive Message (RCVMSG) command is used by a program to receive a message previously sent to a message queue.

The RCVMSG command receives messages from a job message queue (a message queue associated with a call stack entry or the external message queue (*EXT)), or from a named message queue. The program can receive a message from a message queue associated with its own call stack entry or from a message queue associated with another call stack entry.

This command copies a message received in the specified message queue into control language (CL) variables within the program. The message and its attributes are copied into the CL variables specified by the parameters KEYVAR through DTACCSID.

You can specify the message being received by indicating the message type, the reference key of the message, or both. The program receiving the message can also specify, on the RCVMSG command, whether a message is removed from the message queue or left there as an old message. If the specified message queue is not allocated to the job in which this command is entered, or to any other job, the message queue is implicitly allocated by this command for the duration of the command’s processing.

If a message of the specified type does not exist on the queue, the requesting program can either wait for a message to arrive or continue with other processing. This allows a set of message queues to be polled.

If the message received is an unhandled exception message, the program can specify whether this command should handle the exception. An unhandled exception message is an escape, status, or notify message that has been sent to an Integrated Language Environment (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 the ILE procedure can take is to call a CL program that receives the message using this command. More information on actions that can be taken is in the ILE Concepts book, SC41-5606.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGMQ</td>
<td>Call stack entry message queue</td>
<td>Single values: *EXT Other values: Element list</td>
<td>Optional, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Element 1: Relationship</td>
<td>*SAME, *PRV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Call stack entry identifier</td>
<td>Element list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 1: Call stack entry</td>
<td>Character value, *</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Module</td>
<td>Name, *NONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 3: Bound program</td>
<td>Name, *NONE</td>
<td></td>
</tr>
<tr>
<td>MSGQ</td>
<td>Message queue</td>
<td>Single values: *PGMQ Other values: Qualified object name</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Message queue</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 1998, 2006
Call stack entry message queue (PGMQ)

Specifies the call stack entry message queue from which a message is received. The call stack entry message queue can be the *EXT queue or it can be a message queue that is associated with a call stack entry for a program or an ILE procedure.

If values are specified for this parameter, specifying *PGMQ for the Message queue (MSGQ) parameter is allowed.

Single values

*EXT  The message is received from the external message queue of the job. The external message queue is used to communicate with the external requester of the job, such as a display station user.

Element 1: Relationship

Element 1 of this parameter specifies whether the message queue is associated with the program or procedure identified by Element 2, or if it is associated with the caller of the program or procedure.
*SAME
The message is received from the message queue of the program or procedure identified by element 2.

*PRV  The message is received from the message queue of the program or procedure that called the program or procedure identified by element 2 of this parameter.

Note: If the message queue previous to the one identified by element 2 is for an ILE program entry procedure (PEP), the message will be received from the message queue immediately previous to the PEP message queue; effectively this would be two message queues previous to the one identified by element 2.

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

*  Specify 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.

---

**Message queue (MSGQ)**

Specifies the message queue (not a program message queue) from which a message is to be received.

Single values

*PGMQ  The program message queue specified for the Call stack entry message queue (PGMQ) parameter is the only queue from which a message is received.

Qualifier 1: Message queue

name  Specify the name of the message queue from which a message is to be received. If a message queue name is specified, the Call stack entry message queue (PGMQ) parameter cannot be specified.

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.

---

**Message type (MSGTYPE)**

Specifies the type of message received by this program.

*ANY  Any type of message (except a sender’s copy) is received. To receive a sender’s message, MSGTYPE(*COPY) must be specified.
*NEXT
The message that follows the one specified for the Message key (MSGKEY) parameter is received. If there is not another message available, blanks are returned in all CL variables.

When a message is received from a message queue associated with a call stack entry, *NEXT works only for one call stack entry. *NEXT cannot be used to receive messages for multiple call stack entries of the same program.

*PRV
The message previous to the message specified for the Message key (MSGKEY) parameter is received.

*INFO
An informational message is received.

*INQ
An inquiry message is received.

*RPY
A reply message is received. This program has sent an inquiry message to a message queue and expects a reply.

*FIRST
The first message currently on the message queue or program queue is received.

*COPY
A copy of an inquiry message that was previously sent is received by this program. The message queue specified for the Call stack entry message queue (PGMQ) parameter or the Message queue (MSGQ) parameter must be the same queue that was specified for the Message queue to get reply (RPYMSGQ) parameter when the INQ message was sent.

*COMP
A completion message is received. This type of message can only be received from a program message queue.

*DIAG
A diagnostic message is received. This type of message can only be received from a program message queue.

*EXCP
An exception message is received. Exception messages (escape, notify, status) are received by the program in last-in first-out (LIFO) order. The receiving program can monitor for exception messages by using the MONMSG command.

Note: Non-exception messages are received in first-in first-out (FIFO) order.

If an exception message is received from a message queue for a procedure, the related exception may not be handled at the time the RCVMSG command is run. The RMV parameter can be used to specify whether the exception is to be handled by the RCVMSG command.

*RQS
A request message is received. This type of message can only be received from a program message queue.

*LAST
The last message currently on the message queue or program queue is received.

---

**Message key (MSGKEY)**

Specifies the message reference key of the message that is received.

*NONE
No message reference key is specified.

*TOP
The top of the message queue is used. *TOP can be used only when *NEXT is specified for the
Message type (MSGTYPE) parameter. It causes the first message on the message queue to be received. For program message queues, this is the message following the last request message that was received, if any.

Name Specify the name of the CL variable that contains the message reference key of the message to be used by this receive function. The variable must be a character variable having a length of 4 characters.

Wait time (WAIT)

Specifies, in seconds, the length of time that the program waits for a message of the specified type to arrive in the message queue if it is not there when this command is processed. If the message does not arrive in the specified time, the control language (CL) variables named to receive message fields are filled with blanks (or zeros, if they are decimal variables).

The program cannot wait for a message from a program message queue unless it is receiving a reply.

If a wait time is specified (not zero), the message queue is implicitly allocated to the first user whose message is received, and it is not released until the request has been handled by the program.

If a message is sent to a message queue in the same job, and the message queue is in break delivery mode, this parameter is ignored (that implies WAIT(0), which is the default value for the WAIT parameter).

If the value specified for MSGKEY refers to an inquiry message, and MSGTYPE(*RPy) has been specified, the program ignores the WAIT parameter (value for Wait is 0).

0 The program does not wait for the arrival of a message.

*MAX The program waits indefinitely for the arrival of the specified message.

Number-of-seconds Specifies the number of seconds that the program waits for the arrival of a message.

Remove message (RMV)

Specifies whether the message received by the program is removed from the message queue. For messages that are unhandled exceptions, this parameter also specifies whether the exception is to be handled. If *INQ is specified for the Message type (MSGTYPE) parameter, then *NO must also be specified for this parameter so a reply to the inquiry message can be sent, otherwise the default reply will be sent before the unanswered inquiry is removed.

*YES The message is removed from the message queue. If the message is an unhandled exception, the exception is handled by running the RCVMSG command.

*NO The message is not removed from the message queue. It is left on the message queue as an old message. If the message is an unhandled exception, the exception is handled by running the RCVMSG command.

Note: Old messages are messages that have been received but not deleted. An old message can be received again in one of the following ways:

1. The message reference key of the message is specified for the MSGKEY parameter.

2. A message type of *FIRST, *LAST, *NEXT, or *PRV is specified for the Message type (MSGTYPE) parameter.
**KEEPEXCP**
If the message is an exception message and the exception has not been handled, the exception is left unhandled and the message is left on the message queue as a new message. It can be received again by using the RCVMSG command to receive an *EXCP message. If the message is not an exception message, or if it is but the exception has already been handled, the message is left on the message queue as an old message.

To handle an exception after the RCVMSG has been run, the command can be run a second time by specifying RMV(*YES) or RMV(*NO).

**Coded character set ID (CCSID)**
Specifies the coded character set identifier (CCSID) that you want the message text returned in. This only applies to text returned in the MSG, SECLVL and MSGDTA parameters. When replacement data is returned in the MSGDTA parameter or substituted into the text returned in the MSG or SECLVL parameters, only the part of the replacement text that is defined as a character that can be converted (*CHAR) is converted. The rest of the replacement data is not converted. For more information about the *CHAR field, see the ADDMSGD command.

**JOB** The received message is converted to the CCSID of the job before being returned.

**HEX** The received message is not converted before being returned.

**coded-character-set-identifier**
Specify the CCSID that you want your message converted to before being returned. Valid values range from 1 through 65535. See the Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infcnter for a list of valid values. Only CCSID values that a job can be changed to are accepted.

For more information on the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infcnter.

**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. This parameter is only applicable when *YES is specified for the RMV keyword.

**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.
CL var for KEYVAR (4) (KEYVAR)

Specifies the name of the control language (CL) character variable, if any, that contains the message reference key identifying the message received by the program containing this command. At the time the RCVMSG command is processed, the system returns the message reference key to the variable specified by KEYVAR in this command and changes the received message to an old message. The message reference key can then be used in the MSGKEY parameter in a subsequent RCVMSG command to receive the old message. If the message is not found, blanks are returned for the KEYVAR variable. For reply type messages, use the MSGKEY parameter on this command in conjunction with the KEYVAR parameter on the SNDPGMMSG command. The message reference key can also be used by this program for building message subfiles. The CL variable is the name of the field for which the SFLMSGKEY keyword is specified in the DDS for the message subfile.

Note: For message queues not associated with call stack entries, message reference keys can be used again after a message has been received and then removed (by specifying *YES for the RMV parameter).

The variable must be a character variable having a length of 4 characters.

Note: When using the message reference key (obtained from the CL variable specified by the KEYVAR parameter of the Send Program Message (SNDPGMMSG) command) to receive the reply to an inquiry message, note that the message reference key refers to the sender’s copy. The sender’s copy message is located on the reply message queue (which defaults to the program message queue that sent the inquiry message), not the message queue to which the inquiry message was sent.

CL var for 1st level text (MSG)

Specifies the name of the control language (CL) character variable, if any, that contains the message when it is received by the program. This includes the message data fields that were substituted for substitution variables in the text before the message was sent (replies and immediate messages contain no message data fields). This is a variable-length field, but most message text is less than 132 characters in length.

CL var for MSGLEN (5 0) (MSGLEN)

Specifies the name of the control language (CL) decimal variable, if any, that contains the total length of the message text available to be received. The variable must be a decimal variable having a length of 5 positions.

CL var for 2nd level text (SECLVL)

Specifies the name of the CL character variable, if any, that contains the message help received by the program. This includes the message data fields that were substituted for any substitution variables in the text before the message was sent (replies and immediate messages do not have second-level messages). This is a variable-length field, but most online message help is designed to be less than 3000 characters in length.
**CL var for SECLVLLEN (5 0) (SECLVLLEN)**

Specifies the name of the control language (CL) decimal variable, if any, that contains the total length of the message help available to be received. The variable must be a decimal variable having a length of 5 positions.

**CL var for msg data (MSGDTA)**

Specifies the name of the control language (CL) character variable, if any, that contains the message data record received by the program as part of the message. The message data record contains the substitution values (in a single character string) that are used in the text of the received message. The amount of data returned and its format depend on the message. Pointers contained in system messages are invalidated.

**Note:** If you use data that has an invalidated pointer in it an error message can occur.

**CL var for MSGDTALEN (5 0) (MSGDTALEN)**

Specifies the name of the control language (CL) decimal variable, if any, that contains the total length of the message data record available to be received. The variable must be a decimal variable having a length of 5 positions.

**CL var for MSGID (7) (MSGID)**

Specifies the name of the control language (CL) character variable, if any, that contains the message identifier of the message received by the program. If the message being received is an immediate message, the message identifier is returned as blanks. The minimum length of the variable is 7 characters.

**CL var for SEV (2 0) (SEV)**

Specifies the name of the control language (CL) decimal variable, if any, that contains the severity code of the message received by the program. If the message being received is an immediate message, the message severity is not returned. The variable must be a decimal variable having a length of 2 positions.

**CL var for SENDER (80) (SENDER)**

Specifies the name of the control language (CL) character variable, if any, that contains the identification of the sender of the message received through the RCVMSG command. The length of the CL variable depends on the SENDERFMT specification. If SENDERFMT(*SHORT) is specified, the variable must be a minimum of 80 characters. If the CL variable is longer than 80 characters, additional information will be returned. If SENDERFMT(*LONG) is specified, the variable must be a minimum of 720 characters.
Sender format (SENDERFMT)

Specifies which format of the sender identification is returned. This parameter is valid only when the SENDER parameter is specified.

*SHORT

The short format of the sender information is returned. The short format is a minimum of 80 characters. If the CL variable is longer than 80 characters, additional information will be returned. Positions in the CL variable beyond the last returned field will be set to blanks. The following information is returned:

- The first 26 characters identify the sending job
  - Job name (10)
  - User name (10)
  - Job number (6)
- The next 16 characters identify the sending program
  - Program name (12) (for an ILE procedure, this is the bound program name); if the sender type is 3, the first three characters of this field are less than symbols (<<<) followed by the last nine characters of the program name
  - Instruction number (4) (for an ILE procedure, this field is set to blanks)
- The next 13 characters are the date and time
  - Date (7) (in the format 0yymmdd)
  - Time (6) (in the format hhmmss)
- The next 14 characters identify the sent-to call stack entry if the message is sent to a program message queue
  - Program name (10) (for an ILE procedure, this is the bound program name)
  - Instruction number (4) (for an ILE procedure, this field is set to blanks)
- The next 1 character identifies the sender type
  - "0" if the sender is an OPM program or a SLIC program with 12 characters or less
  - "1" if the sender is an ILE procedure and the name is 256 characters or less
  - "2" if the sender is an ILE procedure and the name is more than 256 characters
  - "3" if the sender is a SLIC program with more than 12 characters
- The next 1 character identifies the sent-to type
  - "0" if the receiver is an OPM program
  - "1" if the receiver is an ILE procedure and the name is 256 characters or less
  - "2" if the receiver is an ILE procedure and the name is more than 256 characters
- The next 6 characters are the microseconds
- The last 10 characters are the name of the user profile that the thread was running under when the message was sent, and is returned if the length of the CL variable is at least 87

*LONG

The long format of the sender information is returned. The long format is 720 characters, with the last 30 characters set to blanks. The following information is returned:

- The first 26 characters identify the sending job
  - Job name (10)
  - User name (10)
  - Job number (6)
- The next 13 characters are the date and time
  - Date (7) (in the format 0yymmdd)
  - Time (6) (in the format hhmmss)
• The next 1 character identifies the sender type
  – "0" if the sender is an OPM program or a SLIC program with 12 characters or less
  – "1" if the sender is an ILE procedure and the name is 256 characters or less
  – "2" if the sender is an ILE procedure and the name is more than 256 characters
  – "3" if the sender is a SLIC program with more than 12 characters
• The next 1 character identifies the sent-to type
  – "0" if the receiver is an OPM program
  – "1" if the receiver is an ILE procedure and the name is 256 characters or less
  – "2" if the receiver is an ILE procedure and the name is more than 256 characters
• The next 12 characters are the sender’s program name (for an ILE procedure, this is the bound program name); if the sender type is 3 and the program name is greater than 12 characters in length, the first three characters of this field are less than symbols (<<<) followed by the last nine characters of the program name
• The next 10 characters are the sender’s module name (if the sender is not an ILE procedure, this field is set to blanks)
• The next 256 characters are the sender’s procedure name (if the sender is not an ILE procedure, this field is set to blanks)
  – For a nested procedure name, each procedure name is separated by a colon (:) starting with the outer-most procedure name, and ending with the inner-most procedure name
  – For a procedure name that is longer than 256 characters, three less than symbols (<<<) are returned followed by the last 253 characters of the procedure name; the QMHRCVPM API can be used to obtain the entire procedure name
• The next 1 character is blank
• The next 4 characters are the number of statement numbers available

**Note:** A statement number represents a point in the sending program at which the message was sent. For programs and non-optimized procedures, this count is always 1. For optimized procedures, this count can be greater than 1, and each statement number represents a point at which the message could have been sent. If it is not possible to return statement numbers, this count will be 0.

• The next 30 characters return a maximum of 3 statement numbers, 10 characters each
• The next 320 characters return program or procedure information if the message being received was originally sent to a message queue associated with a call stack entry (otherwise, this field is set to blanks)
  – Sent-to program name (10) (for an ILE procedure, this is the bound program name)
  – Sent-to module name (10) (if the sender is not an ILE procedure, this field is set to blanks)
  – Sent-to procedure name (256) (if the sender is not an ILE procedure, this field is set to all blanks)
    - For a nested procedure name, each procedure name is separated by a colon (:) starting with the outer-most procedure name, and ending with the inner-most procedure name.
    - For a procedure name that is longer than 256 characters, three less than symbols (<<<) are returned followed by the last 253 characters of the procedure name; the QMHRCVPM API can be used to obtain the entire procedure name
  – Blanks (10)
  – Number of statements available for the receiving call stack entry (4)

**Note:** A statement number represents a point at which the sent-to program was suspended (for example, due to a call operation) at the time the message was sent. For programs and non-optimized procedures, this count is always 1. For optimized procedures, this count can be greater than 1, and each statement number represents a point at which the message could have been sent. If it is not possible to return statement numbers, this count will be 0.
• Statement numbers (30) (a maximum of 3 statement numbers, 10 characters each)
• The next 6 characters are the microseconds
• The last 10 characters are the name of the user profile that the thread was running under when the message was sent

#### CL var for RTNTYPE (2) (RTNTYPE)

Specifies the name of the control language (CL) variable, if any, that contains the type code for the message received by the program. The variable must be a character variable having a length of 2 positions.

The following values are returned to indicate the message type:

<table>
<thead>
<tr>
<th>Value</th>
<th>Message Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Completion</td>
</tr>
<tr>
<td>02</td>
<td>Diagnostic</td>
</tr>
<tr>
<td>04</td>
<td>Information</td>
</tr>
<tr>
<td>05</td>
<td>Inquiry</td>
</tr>
<tr>
<td>06</td>
<td>Copy</td>
</tr>
<tr>
<td>08</td>
<td>Request</td>
</tr>
<tr>
<td>10</td>
<td>Request with prompting</td>
</tr>
<tr>
<td>14</td>
<td>Notify (exception already handled at time of RCVMSG)</td>
</tr>
<tr>
<td>15</td>
<td>Escape (exception already handled at time of RCVMSG)</td>
</tr>
<tr>
<td>16</td>
<td>Notify (exception not handled at time of RCVMSG)</td>
</tr>
<tr>
<td>17</td>
<td>Escape (exception not handled at time of RCVMSG)</td>
</tr>
<tr>
<td>21</td>
<td>Reply, not checked for validity</td>
</tr>
<tr>
<td>22</td>
<td>Reply, already checked for validity</td>
</tr>
<tr>
<td>23</td>
<td>Reply, message default used</td>
</tr>
<tr>
<td>24</td>
<td>Reply, system default used</td>
</tr>
<tr>
<td>25</td>
<td>Reply, from System Reply List</td>
</tr>
<tr>
<td>26</td>
<td>Reply, from exit program</td>
</tr>
</tbody>
</table>

#### CL var for ALROPT (9) (ALROPT)

Specifies the name of the control language (CL) variable, if any, used to return the alert option of the message received by the program. The variable must be a character variable 9 positions in length.
**CL var for MSGF (10) (MSGF)**

Specifies the name of the control language (CL) variable, if any, used to return the message file name of the message received by the program. If the message received is a stored message, the message file name of the file containing the stored message is returned. If the received message is not a stored message, the message file name is returned as blanks. The variable must be a character variable 10 positions in length.

**Note:** The message file name returned on this parameter is the message file specified or defaulted on the send function, not the overriding message file. If an override was specified when sending the message, the same override should be used when receiving the message.

---

**CL var for MSGFLIB (10) (MSGFLIB)**

Specifies the name of the control language (CL) variable, if any, used to return the message file library name of the message received by the program. If the message received is a stored message, the message file library name of the library containing the message file of the stored message is returned. If *LIBL was specified on the send command, *LIBL is returned. If the received message is not a stored message, the message file library name is returned as blanks. The variable must be a character variable 10 positions in length.

**Note:** The message file library name returned on this parameter is the message file specified or defaulted on the send function, not the overriding message file library. If an override was specified when sending the message, the same override should be used when receiving the message.

---

**CL var for SNDMSGFLIB (10) (SNDMSGFLIB)**

Specifies the name of the control language (CL) variable, if any, used to return the message file library name used to send the message. If the message received is a stored message, the message file library name of the library containing the message file of the stored message is returned. If *LIBL was specified on the send command, this parameter would have the actual name of the library. If the received message is not a stored message, the message file library name is returned as blanks. If the received message is a stored message and the original message file is destroyed, the message file library name is returned as blanks. If the received message is not a stored message, this parameter is returned as blanks. The variable must be a character variable of 10 positions in length.

---

**CL var for text CCSID (5 0) (TXTCCSID)**

Specifies the name of the CL variable, if any, used to return the coded character set identifier (CCSID) associated with the text returned by the MSG and SECLNL parameters. If a conversion error occurs or if the CCSID you requested the text to be converted to is 65535, the CCSID that the message description or the text for an immediate message is stored in is returned. Otherwise, the CCSID you wanted the text converted to is returned. If you do not want the text converted before it is returned to you but you do want to know the CCSID that the message description or the text for an immediate message is stored in, specify 65535 for the CCSID parameter, and the CCSID is returned in the TXTCCSID parameter. You can also check for a conversion error by comparing the CCSID you passed in against the TXTCCSID returned. If they are not equal and they are not 65535, a conversion error occurred. The variable must be a decimal variable having a length of 5 positions.
**CL var for data CCSID (5 0) (DTACCSID)**

Specifies the name of the CL variable, if any, used to return the coded character set identifier (CCSID) associated with the replacement data defined as *CCHAR. All other replacement data is not converted before it is returned. If a conversion error occurs or if the CCSID you requested the text to be converted to is 65535, the CCSID message data is returned. If there is no *CCHAR replacement data in the data, 65535 is returned. Otherwise, the CCSID you wanted the text converted to is returned. For immediate messages, 0 is returned. You can check for a conversion error by comparing the CCSID you passed in against the DTACCSID returned. If they are not equal and they are not 65535, a conversion error occurred. The variable must be a decimal variable having a length of 5 positions.

**Examples**

**Example 1: Receiving a Message**

RCVMSG MSGQ(SMITH) MSGKEY(&KEY) MSG(&WORK)

This command receives the message having the message reference key specified by the program variable &KEY from the message queue SMITH. The text of the message is copied into the CL variable &WORK.

**Example 2: Receiving a New Message**

RCVMSG MSGQ(INV) WAIT(120) MSG(&WORK)

This command receives a new message from the message queue named INV into the CL variable &WORK. The program waits no more than 120 seconds for the arrival of a new message if there are no new messages in the message queue. If there is more than one new message in the queue, the first message in the queue is the message received by the program.

**Example 3: Receiving a Message From a Procedure**

RCVMSG PGMQ(*SAME CURRENT_MONTH_TOTALS) MSGTYPE(*EXCP) RMV(*KEEPEXCP) MSGID(&MID) MSG(&MTEXT)

This command receives an exception message from the procedure CURRENT_MONTH_TOTALS. Since the specified name is more than 10 characters, the system does not search for any programs. If the message is an unhandled exception, the message is left on the call message queue as a new message and the exception is not handled by the RCVMSG command. The message ID is returned in the CL variable &MID and the message text in the CL variable &MTEXT. To handle the exception and remove the message, run the following RCVMSG command:

RCVMSG PGMQ(*SAME CURRENT_MONTH_TOTALS) MSGTYPE(*EXCP) RMV(*YES)

**Example 4: Receiving a Message from a Program or Procedure**

RCVMSG PGMQ(*SAME TARGETPGM) MSGTYPE(*EXCP) RMV(*NO) MSGID(&MID) MSG(&MTEXT)

This command receives an exception message from the message queue of the program or procedure named TARGETPGM. Since the specified name is only 9 characters, the system searches both programs and procedures. Because RMV(*NO) is specified, if the message is an unhandled exception, the exception is handled by the RCVMSG command. The message is left on the message queue as an old message.

**Example 5: Receiving a Message Using Qualifiers**
This command receives an exception message from the message queue of the procedure named PRINT_RPT_FMT1. The procedure must have been compiled into the module DEPTRPTS and have been bound into the bound program AREARPTS. Since RMV(*YES) is specified, the exception is handled if the exception message is for an unhandled exception. The message is always removed from the message queue.

Example 6: Receiving a Message Using a Partial Procedure Name

```
RCVMSG PGMQ(*SAME 'HANDLE_FORM_NUM>>>') MSGID(&MID)
       MSG(&MTEXT)
```

This command receives a new message from the most recent procedure whose name begins with HANDLE_FORM_NUM.

---

### Error messages

***ESCAPE Messages**

- **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.

- **CPF2411**
  - Not authorized to message file &1 in &2.

- **CPF241C**
  - Variable for SENDER parameter is too small.

- **CPF2410**
  - Message key not found in message queue &1.

- **CPF2415**
  - End of requests.

- **CPF2422**
  - Reply not valid.

- **CPF2423**
  - Variable specified in SENDER parameter less than &1 bytes.

- **CPF2433**
  - Function not allowed for system log message queue &1.

- **CPF2449**
  - Message that should be a reply, is not a reply.

- **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.

CPF247E
CCSID &1 is not valid.

CPF247I
Length of field not valid.

CPF2477
Message queue &1 currently in use.

CPF2479
Call stack entry not found.

CPF2482
Message type &1 not valid.

CPF24A3
Value for call stack counter parameter not valid.

CPF24A8
Value for wait time not valid.

CPF24B3
Message type &1 not valid.

CPF2531
Message file &1 in &2 damaged for &3.

CPF2532
Job message queue is damaged. Job log ended.

CPF2548
Damage to message file &1 in &2.

CPF2551
Message key and message type combination not valid.

CPF36F7
Message queue QSYSOPR is allocated to another job.

CPF8127
&8 damage on message queue &4 in &9. VLIC log-&7.

CPF8176
Message queue for device description &4 damaged.
Receive Network File (RCVNETF)

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

The Receive Network File (RCVNETF) command receives a network file and copies the records into a physical database file or a save file. Once the file has been received, it is removed from the queue of network files.

If the original file is a save file, it must be received into a save file. Before a file can be received, the file specified by the TOFILE parameter must already exist.

When a source physical file is sent, the source sequence number and change date in positions 1 through 12 of the record are sent with the file. These are kept if the file is received into a source physical file, and are truncated if the file is received into a nonsource physical file. When a file that was originally a nonsource physical file is received into a source physical file, the source sequence numbers are created and placed in front of the records.

If the file is a physical file, the record length of the to-file must be at least as large as the record length of the original file. If the record length of the to-file is larger than that of the original file, the records are padded to the end with the default record value for the to-file.

This command does not perform any CCSID translation on the contents of the file. However, the user ID and address of both the recipient and the originator are translated from the multinational character set 697/500 to the current job CCSID.

Restrictions:
1. A user with security officer authority can receive the files sent to any user. Users with other than security officer authority can receive only files sent to them or to their group profile.
2. The user must have read authority to the library containing the to-file, and use and add authority to the to-file. The following additional authority may be required:
   • Object management authority, if a member is added to the file.
   • Object management authority and delete authority, if a save file or existing physical file member is cleared.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROMFILE</td>
<td>From file</td>
<td>Character value</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>TOFILE</td>
<td>To file</td>
<td>Qualified object name</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: To file</td>
<td>Name, *FROMFILE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>FROMMBR</td>
<td>Member to be received</td>
<td>Character value, *ONLY</td>
<td>Optional, Positional 3</td>
</tr>
<tr>
<td>TOMBR</td>
<td>To member</td>
<td>Name, *FROMMBR, *FIRST</td>
<td>Optional, Positional 4</td>
</tr>
</tbody>
</table>
### MBROPT

Replace or add records

**Choices**
- *REPLACE
- *ADD

**Notes**
Optional

### NBR

File number

**Choices**
- Integer
- *LAST
- *ONLY
- *FIRST

**Notes**
Optional

### USER

User

**Choices**
- Name
- *CURRENT

**Notes**
Optional

### FROMTYPE

From file type

**Choices**
- *NETFILE
- *SRC

**Notes**
Optional

---

**From file (FROMFILE)**

Specifies the name of the file that is received. This is the name of the file on the sending system.

This is a required parameter.

---

**To file (TOFILE)**

Specifies the name of the file into which the network file is received. Overrides to this file are ignored.

The possible values are:

*FROMFILE

The network file is received into a file of the same name as the file sent.

**file-name**

Specify the name and library of the receiving file.

The possible library values are:

*LIBLE

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.

---

**Member to be received (FROMMMBR)**

Specifies the name of the file member that is received.

The possible values are:

*ONLY

Only one member is received for this file. If *ONLY is specified on the **File number** prompt (NBR parameter), only one member per file is on the arrived file queue.

**member-name**

Specify the name of the member that is received. A member name cannot be specified if the file is a save file.
To member (TOMBR)
Specifies the database file member that receives the data.

The possible values are:

*FROMMBR
The data is received into a member with the same name as the member specified on the Member to be received prompt (FROMMBR parameter).

*FIRST
The first member in the file receives the output.

member-name
Specify the name of the member that receives the records. A member name cannot be specified if the file is a save file.

Replace or add records (MBROPT)
Specifies whether the new records replace or are added to the existing records.

The possible values are:

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

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

File number (NBR)
Specifies the number of the file member that is received. This number is used to identify the member that is received when there is more than one member of the same name available for the file.

The possible values are:

*LAST
The last network file member with the specified member name is received. The last member is determined as the last member to arrive at your system.

Note: The file member that arrived last at the user’s system may not have been the last one sent by the sending user. The network does not guarantee the arrival sequence of separately sent files.

*ONLY
Only one file member of the specified file name is received.

member-number
Specify the number of the member that is received.
**User (USER)**

Specifies the user to whom the file was sent.

The possible values are:

*CURRENT

The files sent to the current user are received.

user-name

Specify the name of the user to whom the files were sent. Only users with security officer authority can specify a name other than their own or their group profile.

---

**From file type (FROMTYPE)**

Specifies the type of file that is received. This option should be used mainly when the file is an AS/400 or System/38 source file which was sent by a System/370 VM or MVS user. Since VM or MVS cannot identify whether the file is a source file, you can indicate whether the file is a source file or a non-source file.

The possible values are:

*NETFILE

The network file type is used to determine whether file type conversion is needed.

If the file is a non-source file and is:

- Received into a non-source file, the file is received unchanged.
- Received into a source file, the sequence numbers and date fields are added.

If the file is a source file and is:

- Received into a non-source file, the sequence numbers and date fields are removed (the first 12 bytes of each record).
- Received into a source file, the file is received unchanged.

*SRC

The file being received is a source file. The sequence numbers and date fields are in the file. If the file is received into another source file, the sequence numbers and date fields are not added to the file being received. If the file is received into a non-source file, the sequence numbers and date fields are removed (the first 12 bytes of each record).

**Note:** *SRC must not be specified if the network file does not contain sequence numbers and date fields in the first 12 bytes of each record.

---

**Examples**

**Example 1: Receiving a Member**

```
RCVNET FROMFILE(FILEA) TOFILE(FILEB/FILEA) FROMMBR(PAYROLL)
```

This command receives member PAYROLL of file FILEA into member PAYROLL of file FILEA in library FILEB. If there is an existing member of that name, the records in the member are replaced. If multiple members of that name are available, the last one to arrive at the destination system is received.

**Example 2: Receiving a Network File**
RCVNETF FROMFILE(PERSONNEL) NBR(*LAST) USER(USR1)

This command receives a network file named PERSONNEL, which was sent to user USR1, into a file with the same name. Because the FROMMMBR parameter is not specified, there must be only one member name available for this file. Because USR1 is specified, only someone with a user profile of USR1, someone with a group profile of USR1, or someone with security officer authority can use this command.

Example 3: Receiving a Source File
RCVNETF FROMFILE(FILEA) TOFILE(FILEB/FILEA)
FROMMMBR(PAYROLL) FROMTYPE(*SRC)

This command specifies that the file being received is a source file and the sequence numbers and date fields are not added to the file being received.

Error messages
*ESCAPE Messages
CPF2204 User profile &1 not found.
CPF2207 Not authorized to use object &1 in library &3 type *&2.
CPF4101 File &2 in library &3 not found or inline data file missing.
CPF5715 File &1 in library &2 not found.
CPF8057 File &1 in &2 not a physical file or save file.
CPF8059 Member name not allowed for save file.
CPF8060 No files compare to the specified selection.
CPF8062 Record length of network file larger than receiving file.
CPF8063 Cannot assign necessary resource.
CPF8070 Not allowed to process files for user &1.
CPF8077 More than one file with same name found. See previously displayed messages.
CPF8080 MBROPT(*ADD) not allowed for save file.
CPF8081 File &5 member &6 number &7 already processed.
CPF8082 Cannot get network file &5 member &6 number &7.
CPF9005
System resource required to complete this request not available.

CPF9006
User not enrolled in system distribution directory.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

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

CPF9830
Cannot assign library &1.

CPF9843
Object &1 in library &3 type &2 cannot be accessed.

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.

CPF9848
Cannot open file &1 in library &2 member &3.

CPF9849
Error while processing file &1 in library &2 member &3.
Receive TIE File (RCVTIEF)

Where allowed to run:
- Batch job (*BATCH)
- Batch program (*BPGM)
- Batch REXX procedure (*BREXX)
- Using QCMDEXEC, QCAEXEC, or QCPCMD API (*EXEC)

Threadsafe: No

The Receive Technical Information Exchange File (RCVTIEF) command allows you to receive files transmitted from the remote support network.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIB</td>
<td>Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>TYPE</td>
<td>File type</td>
<td>*ALL, *OTHER, *SAVF</td>
<td>Optional</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Output</td>
<td>*NONE, *PRINT</td>
<td>Optional</td>
</tr>
<tr>
<td>MAXRCDS</td>
<td>Maximum records</td>
<td>Integer, 10000, *NOMAX</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Library (LIB)

Specifies the name and library where the files are stored.

This is a required parameter.

The possible library values are:
*LIBL  The library list is used to locate the database file.
*CURLIB  The current library for the job is used to locate the database file. If no library is specified as the current library for the job, QGPL is used.

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

File type (TYPE)

Specifies the types of files that are received.

*ALL  All available files are received.
*OTHER
Files with unspecified contents are received.

*SAVF  Save files are received.

Output (OUTPUT)
Specifies whether the output from the command is displayed at the requesting work station or printed with the job’s spooled output.

*NONE
The list of received files is not printed.

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

Maximum records (MAXRCDS)
Specifies the maximum size (number of records) of any file that can be received. If one or more files exceeds the maximum size, no files are received.

10000  The maximum size of file that can be received is 10000 records.

*NOMAX
The system maximum of 500000 records is used.

number
Specify the maximum size (number of records) of file that can be received.

Examples

RCVTIEF LIB(MAIL) TYPE(*OPEN) OUTPUT(*PRINT) MAXRCDS(1000)

This command receives from TIE all OPEN files (any file except a save file). A list of the received files is printed. If any of the received files are larger than 1000 records, the RCVTIEF command fails. If all OPEN files are received successfully, they are removed from the mailbox.

Error messages
None
Remove Directory (RD)

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

The Remove Directory (RD) 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 is an alias for the Remove Directory (RMVDIR) command and can also be issued using the following alternative command names:

- RMDIR
- RMVDIR

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.
**Parameters**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR</td>
<td>Directory</td>
<td>Path name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>RMVLNK</td>
<td>Remove link</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**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/iseries/infocenter.

**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.

**Examples**

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

**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.
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

CPFA0B2
No objects satisfy request.

CPFA0B7
&1 directories removed. &2 directories not removed.
Rename Object (REN)

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

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

This command is an alias for the Rename Object (RNM) command and can also be issued using the following alternative command names:

- RNM

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. 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.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ</td>
<td>Object</td>
<td>Path name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>NEWOBJ</td>
<td>New object</td>
<td>Character value</td>
<td>Required, Positional 2</td>
</tr>
</tbody>
</table>
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/iseries/infocenter.

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.

Examples

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

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.

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

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.
Return (RETURN)

Where allowed to run:
- Interactive job (*INTERACT)
- Batch program (*BPGM)
- Interactive program (*IPGM)
- Interactive REXX procedure (*IREXX)

Threadsafe: Yes

The Return (RETURN) command returns control either to the next higher call stack entry in the call stack, or to the subsystem monitor that controls the job.

When used outside a CL program, this command performs the same function as the F3 key. It returns control from the most recent invocation of QCMD (the IBM-supplied control language processor that interprets and processes CL commands for the system) back to the outside program manager. When used in a CL program, this command returns control to the next command or high-level language statement in the calling program at the point where it called the returning program. If this command is used in the highest invocation level in the routing step (either the QCMD program, which is the interpretive CL command processor, or a CL program), the routing step is ended.

Note: If the RETURN command is entered interactively from the highest recursion level while the subsystem is undergoing a controlled end resulting from
- An End Subsystem (ENDSBS) command
- An End System (ENDSYS) command
- A Power Down System (PWRDWN) command

end-of-job processing occurs unless you receive the inquiry message and indicate that you want to return to the command entry display.

There are no parameters for this command.

Parameters

None

Examples

RETURN

When used in a CL program, this command returns control to the CL command or high-level language statement immediately following the point in the last calling program at which this program was called. When used in an interactive job, this command returns control to the next higher level of QCMD. If the RETURN command is run in the highest call level program (QCMD) in the routing step, an inquiry message is sent, and the user has the option of returning to the command entry display. Otherwise, the routing step ends as usual.
Error messages

*ESCAPE Messages

None.
Reorganize Document Lib Object (RGZDLO)

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

The Reorganize Document Library Object (RGZDLO) command allows you to reorganize:
- all document library objects (optionally including unfiled mail documents)
- all filed documents not contained in a folder
- all document library objects within a specified folder
- all unfiled mail documents
- an individual document library object, specified by folder name, document name, or system-object-name

When a document is reorganized, unused storage is removed.

Restrictions:
- You must have all object (*ALLOBJ) or security administrator (*SECADM) special authority to specify DLO(*ALL) with either FLR(*ANY) or FLR(*NONE).
- To reorganize a document or folder, you must have *ALLOBJ or *SECADM special authority or you must have at least change (*CHANGE) authority to the document or folder and be enrolled in the system directory.
- To reorganize a document or folder, you must have exclusive access.
- To reorganize unfiled mail documents, all document and folder activity on the system must be stopped.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLO</td>
<td>Document library</td>
<td>Character value, *ALL, *SYSOBJNAM</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>FLR</td>
<td>Folder</td>
<td>Character value, *NONE, *ANY</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>SYSOBJNAM</td>
<td>System object</td>
<td>Name</td>
<td>Optional</td>
</tr>
<tr>
<td>DAYS</td>
<td>Days since last</td>
<td>0-999, 0</td>
<td>Optional</td>
</tr>
<tr>
<td>MAIL</td>
<td>Reorganize unfiled</td>
<td>*NO, *YES, *ONLY</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Document library object (DLO)

 Specifies the name of the object being reorganized.

This is a required parameter.

*ALL All document library objects are reorganized. If FLR(*NONE) is specified with this parameter, all folderless documents are reorganized. If FLR(*ANY) is specified with this parameter, all filed
documents and folders are reorganized. If MAIL(*YES) is specified with this parameter, all unfiled mail documents as well as all filed documents and folders are reorganized. If MAIL(*ONLY) is specified with this parameter, only unfiled mail documents are reorganized. If FLR(name) is specified with this parameter, all folders and documents within it are reorganized.

*SYSOBJNAM
A system object name specified on the System object name (SYSOBJNAM) parameter is used to identify the document or folder being reorganized.

name Specify the name of the document or folder to be reorganized. The Folder (FLR) parameter also can be used to reorganize a document by specifying reorganization of:
- The folder that contains the document being reorganized
- The folder that contains the nested folder that contains the document being reorganized

Folder (FLR)
Specifies the folder that contains the documents or folders. If the documents or folders do not exist in a folder, *NONE is specified.

*NONE
The document or folder is not contained in a folder. When DLO(*ALL) is specified, this refers to all documents not contained in a folder; when DLO(name) is specified, this refers to a first-level folder.

*ANY
All document library objects will be reorganized including those not contained in any folder. This value is valid only when DLO(*ALL) is specified.

name Specify the name of the folder that contains the documents or folders.

System object name (SYSOBJNAM)
Specifies the system object name of an individual folder or document being reorganized.

Days since last referenced (DAYS)
Specifies the number of days that must have elapsed since a document library object was last referred to before it can be reorganized.

0
All objects requested will be reorganized.

1-999
Specify the number of days that must have elapsed since a document library object was referred to before it can be reorganized.

Reorganize unfiled mail (MAIL)
Specifies whether objects to be reorganized should include, omit, or be limited to unfiled mail documents.

*NO
Unfiled mail documents will not be reorganized.
*YES  Unfiled mail documents will be reorganized. This value is valid only when DLO(*ALL) and FLR(*ANY) are specified.

*ONLY  Only unfiled mail documents will be reorganized. This value is valid only if DLO(*ALL) and FLR(*ANY) are specified.

**Examples**

Example 1: Reorganizing Folders and Documents

`RGZDLO DLO(*ALL) FLR(*ANY)`

This command reorganizes all filed folders and documents that exist on the system.

Example 2: Reorganizing Folders, Documents, and Unfiled Mail

`RGZDLO DLO(*ALL) FLR(*ANY) MAIL(*YES)`

This command reorganizes all filed folders, documents, and all unfiled mail documents that exist on the system.

Example 3: Reorganizing Unfiled Mail Documents

`RGZDLO DLO(*ALL) FLR(*ANY) MAIL(*ONLY)`

This command reorganizes all unfiled mail documents that exist on the system.

Example 4: Reorganizing Folderless Documents

`RGZDLO DLO(*ALL) FLR(*NONE)`

This command reorganizes all folderless documents that exist on the system.

Example 5: Reorganizing Documents Within Folders Within Folders

`RGZDLO DLO(*ALL) FLR(FLRA)`

This command reorganizes all documents within folders contained in folder FLRA, then the folders within folder FLRA are reorganized.

Example 6: Reorganizing an Individual Document or Folder

`RGZDLO DLO(*SYSOBJNAM) SYSOBJNAM(DCN1371951)`

This command reorganizes the individual document or folder identified by the SYSOBJNAM object.

Example 7: Reorganizing a Document

`RGZDLO DLO(DOC1) FLR(FLRA)`

This command reorganizes the document named DOC1 in folder FLRA.

Example 8: Reorganizing Documents Not Referenced

`RGZDLO DLO(*ALL) FLR(*ANY) DAYS(30)`

This command reorganizes all filed documents and folders that have not been referenced in the past 30 days.
Error messages

*ESCAPE Messages

CPF8AB1
&1 objects reorganized; &2 objects not reorganized.

CPF8AB2
RGZDLO command failed.
Reorganize Physical File Mbr (RGZPFM)

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

The Reorganize Physical File Member (RGZPFM) command removes deleted records from (compresses) one member of a physical file in the database, and it optionally reorganizes that member.

If a keyed file is identified in the Key file (KEYFILE) parameter, the system reorganizes the member by changing the physical sequence of the records in storage to either match the keyed sequence of the physical file member’s access path, or to match the access path of a logical file member that is defined over the physical file. Reorganization can decrease file processing time when a program is reading sequentially through a keyed physical file or through a keyed logical file.

When the member is reorganized and KEYFILE(*NONE) is not specified, the sequence in which the records are actually stored is changed, and any deleted records are removed from the file. If KEYFILE(*NONE) is specified or defaulted, the sequence of the records does not change, but deleted records are removed from the member. Optionally, new sequence numbers and zero date fields are placed in the source fields of the records. These fields are changed after the member has been compressed or reorganized.

Notes:
• If you cancel this command, the system rebuilds any access paths that are not maintained during the reorganize. If ALWCANCEL(*NO) is specified, any updates to a physical file member with a unique access path over it are prevented until the access path is completely rebuilt.
• If you cancel this command and ALWCANCEL(*YES) is specified, the reorganize will be partially complete. Subsequently, another reorganize with the same parameters may be able to continue from where the last reorganize ended.
• The RGZPFM command ignores all file overrides that are currently in effect for the job. The file names specified on the FILE and KEYFILE parameters identify the files actually used in the reorganize operation, regardless of overrides that may exist for these files.

Restrictions:
• During the reorganization of a physical member, the file being reorganized is locked. The Lock state (LOCK) parameter can be specified to indicate how much concurrent access to the physical file member should be allowed.
• The user needs object operational authority, object management or alter authority, all data authority to the physical file containing the member to be reorganized, and execute authority to the library. The user also needs object operational authority to the file specified on the KEYFILE parameter and execute authority to the library.
• In multi-threaded 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.
Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>Data base file</td>
<td>Qualified object name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Data base file</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>MBR</td>
<td>Member</td>
<td>Name, *FIRST, *LAST</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td>KEYFILE</td>
<td>Key file</td>
<td>Single values: *NONE, *FILE, *RPLDLTRCD, Other values: Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Logical file</td>
<td>Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Logical file</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Member</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>RBDACCPTH</td>
<td>Rebuild access paths</td>
<td>*YES, *OPTIMIZE, *NO</td>
<td>Optional</td>
</tr>
<tr>
<td>ALWCANCEL</td>
<td>Allow cancel</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
<tr>
<td>LOCK</td>
<td>Lock state</td>
<td>*EXCL, *EXCLRD, *SHRUPD</td>
<td>Optional</td>
</tr>
<tr>
<td>SRCOPT</td>
<td>Source update options</td>
<td>Single values: *SAME, Other values (up to 2 repetitions): *SEQNBR, *DATE</td>
<td>Optional</td>
</tr>
<tr>
<td>SRCSEQ</td>
<td>Source sequence numbering</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Starting sequence number</td>
<td>0.01-9999.99, 1.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Increment number</td>
<td>0.01-9999.99, 1.00</td>
<td></td>
</tr>
<tr>
<td>RCDFMT</td>
<td>Record format</td>
<td>Name, *ONLY</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Data base file (FILE)

Specifies the physical file whose member is to be reorganized.

This is a required parameter.

Qualifier 1: Data base 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 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.

Member (MBR)

Specifies the member to be reorganized.
*FIRST

The first, or the only, member in the file is reorganized.

*LAST

The last, or the only, member in the file is reorganized.

name

Specify the name of the file member to be reorganized.

---

**Key file (KEYFILE)**

Specifies whether the physical file member has its arrival sequence changed to match its keyed sequence, is reorganized in the sequence of a logical file member, or is not reorganized. If this parameter specifies a multiple-format logical file and member, the **Record format (RCDFMT)** parameter must also be specified.

**Note:** Join logical files cannot be specified as key files, and a logical file in this parameter is not allowed to have a select/omit access path.

**Single values**

**NONE**

The member is not reorganized; it is only compressed by having deleted records removed.

**RPLDLTRCD**

The member will be reorganized by replacing deleted records at the start of the file with valid records from the end of the file. If the rows must exactly match the current arrival sequence, do not use *RPLDLTRCD.

**FILE**

For a physical file member having a keyed sequence access path, the arrival sequence of the records in the member is changed to match their keyed sequence.

**Element 1: Logical file**

Specify the name and library of the logical file associated with the physical file member.

**Qualifier 1: Logical file**

*file-name*

Specify the name of the logical file to use as the key file.

**Qualifier 2: Library**

*LIBL*

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

*CURLIB*

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

*library-name*

Specify the name of the library to be searched.

**Element 2: Member**
**name** Specify the member of the logical file whose sequence is used to reorganize the physical file member.

---

**Rebuild access paths (RBDACCPTH)**

Specifies whether to rebuild or maintain any valid access paths (other than an access path specified as the KEYFILE or a MAINT(*REBLD) access path) over the member.

- ***YES** Access paths will be rebuilt synchronously at the end of the reorganize operation.
- ***OPTIMIZE**
  Access paths will either be rebuilt asynchronously at the end of the reorganize operation, or maintained during the reorganize, based on which method will result in the access paths being rebuilt the fastest.
- ***NO** Access paths will be maintained during the reorganize.
  If *NO is specified, ALWCANCEL(*YES) must also be specified.

---

**Allow cancel (ALWCANCEL)**

Specifies whether to allow the reorganize to be canceled.

- ***NO** The reorganize cannot be canceled. If the job ends and the reorganize of the data has not finished, any changes up to the point of the failure are discarded.
  If *NO is specified, KEYFILE(*RPLDLTRCD) and RDBACCPTH(*NO) must not be specified, and LOCK(*EXCL) must also be specified.
- ***YES** The reorganize can be canceled. If a reorganize is canceled before it is finished, a subsequent reorganize with the same parameters will typically continue the reorganize from where it ended. If the number of changes that have occurred since the reorganize was canceled is too large, the reorganize may be restarted rather than continued.

---

**Lock state (LOCK)**

Specifies the lock to acquire on the physical file member. The lock state will determine how much concurrent access is allowed by other jobs during the reorganize.

- ***EXCL**
  An exclusive lock is acquired. No concurrent access to the physical file member is allowed from another job.
- ***EXCLRD**
  An exclusive allow read lock is acquired. Concurrent read access to the physical file member is allowed from another job.
  If *EXCLRD is specified, ALWCANCEL(*YES) must also be specified.
- ***SHRUPD**
  A shared update lock is acquired. Concurrent update, delete, and insert access to the physical file member is allowed from another job.
If *SHRUPD is chosen, the resulting order of the rows may not exactly match what was requested on the KEYFILE keyword. The rows will be reorganized to closely match the specified order, but concurrent update, delete, and insert operations will cause some rows to not be reorganized. If the rows must exactly match the arrival sequence (*NONE) or a keyed file sequence, do not use *SHRUPD.

If *SHRUPD is specified, ALWCANCEL(*YES) must also be specified.

Note: Additional details about lock states can be found in the CL information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Source update options (SRCOPT)

Specifies, for physical source files only, whether the member places new numbers in the sequence number field, places zeros in the date field, or changes both fields. Changes occur after the records are compressed or reorganized.

Single values

*SAME

The sequence number field and date field of records do not change.

Other values

*SEQNBR

The records have a new sequence number placed into the sequence number field. The SRCSEQ parameter specifies a start value and a value to be added.

*DATE

The records have a null date (000000) placed in the date field.

Source sequence numbering (SRCSEQ)

Specifies, only when *SEQNBR is also specified on the SRCOPT parameter, the sequence number that is given to the first record in the source file member and the increment value that is used to renumber all other records in the member. If the member is renumbered but this parameter is not specified, 1.00 is assumed for sequence number and increment value.

Element 1: Starting sequence number

1.00  The first source record in the member has a sequence number of 0001.00.

0.01-9999.99

Specify the sequence number of the first source record in the member. A whole number of up to 4 digits or a fraction of up to 2 digits can be specified. If the starting value contains a fraction, a decimal point must be used.

Element 2: Increment number

1.00  The source records are renumbered in the member with whole number increments of 1 (for example, 1.00, 2.00, 3.00...).

0.01-9999.99

Specify the increment value for renumbering all source records following the first record. A whole
number of no more than four digits or a fraction of no more than two digits can be specified. If the increment value contains a fraction, a decimal point must be used.

For example, if SRCSEQ(5000 10) is specified, the first record in the reorganized member is numbered 5000.00, the second is 5010.00, the third is 5020.00, and so on. If SRCSEQ(*N .25) is specified, the records are numbered 1.00, 1.25, 1.50, 1.75, 2.00, and so on. If a starting value of .01 and an increment value of .01 are specified, there are 999,999 unique sequence numbers possible. If the maximum sequence number of 9999.99 is reached, the remaining records are also assigned the sequence number 9999.99.

Record format (RCDFMT)

Specifies the record format name if the physical file member is reorganized in the sequence of a multiple-format logical file.

*ONLY

The logical file specified by the Key file (KEYFILE) parameter has only one record format. That format is used to reorganize the physical file member.

name

Specify the name of a record format in the multiple-format logical file that is used to reorganize the physical file member.

Examples

Example 1: Reorganizing by Deleting Records

RGZPFM FILE(PAYROLL) MBR(MBR1)

This command compresses member MBR1 of the PAYROLL file by removing the deleted records from the file member.

Example 2: Reorganizing by Replacing Deleted Records

RGZPFM FILE(PAYROLL) MBR(MBR1) KEYFILE(*RPLDLTRCD) ALWCANCEL(*YES) LOCK(*EXCLRD)

This command compresses member MBR1 of the PAYROLL file by replacing deleted records at the start of the file with valid records from the end of the file. The command may be canceled and other jobs are allowed to read data from the PAYROLL file while the reorganize is in progress.

Example 3: Reorganizing in Keyed Sequence

RGZPFM FILE(QCLSRC) MBR(CLMBR2) SRCOPT(*SEQNBR *DATE) KEYFILE(*FILE) SRCSEQ(1.00 .25)

This command reorganizes the member CLMBR2 of the CL source file QCLSRC in keyed sequence, with the sequence number field used as the key. The reorganized member has new sequence numbers (starting at 1.00 and incrementing by .25) and a null date (000000) placed in all records when the original member is reorganized.

Error messages

*ESCAPE Messages
CPF2981
Member &3 file &1 in &2 not reorganized.

CPF3135
Access path for member &2 already in use.

CPF32B8
Distributed file error, reason code &3.

CPF32CF
Distributed file error, reason code &3.

CPF32C3
Distributed file error, level ID mismatch

CPF9801
Object &2 in library &3 not found.

CPF9809
Library &1 cannot be accessed.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

*NOTIFY Messages

CPF2985
Source sequence numbers exceeded maximum value allowed. (G C)
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPHT (Merge TCP/IP Host Table)
Release Communications Device (RLSCMNDEV)

Where allowed to run: All environments (*ALL)

Threadsafe: No

The Release Communications Device (RLSCMNDEV) command restores the communications capability of a specified device held by the Hold Communications Device (HLDCMNDEV) command.

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

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV</td>
<td>Device</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
</tbody>
</table>

Device (DEV)

Specifies the name of the device whose communications are released after being held. Specify the name of the device. Devices whose communications can be held by the HLDCMNDEV command are:

DEV Value

Device

3180 Display station
3277 Display station
3278 Display station
3279 Display station
3287 Printer (work station)
5219 Printer (work station)
5224 Printer (work station)
5225 Printer (work station)
5251 Display station
5252 Display station
5256 Printer (work station)
5291 Display station
5292 Display station
PLU1 Primary logical unit, type 1 (for SNA)
BSC Binary synchronous device (Base and RJE)
This &sys. system is a BSC multipoint tributary station

Logical unit in advanced program-to-program communications network

This is a required parameter.

Examples

RLSCMNDEV  DEV(WSPR05)

This command restores the communications capability of the currently held device WSPR05.

Error messages

*ESCAPE Messages

CPF5920
Device &1 varied off or in diagnostic mode.

CPF5921
Device &1 not a communications device.

CPF5935
Error occurred during command processing.

CPF5984
Not authorized to perform function.

CPF9814
Device &1 not found.

CPF9825
Not authorized to perform function.
Release Distribution Queue (RLSDSTQ)

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

The Release Distribution Queue (RLSDSTQ) command releases a distribution queue from held status and allows it to be sent.

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

Restrictions:
- This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
- Messages that report errors about distribution queues may display or print different characters than you entered for the distribution queue name because of internal system transformations. Similarly (depending on the language used for the work station), the internal value for a distribution queue name may differ from the characters shown for the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the Distribution queue prompt (DSTQ parameter) does not match the rules for an internal distribution queue value or if it does not match the internal value for any defined distribution queue (ignoring case differences).

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSTQ</td>
<td>Distribution queue</td>
<td>Character value</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>PTY</td>
<td>Priority</td>
<td>*NORMAL, *HIGH</td>
<td>Required, Positional 2</td>
</tr>
</tbody>
</table>

Distribution queue (DSTQ)

Specifies the name of the distribution queue that is released. The queue must have been previously configured using the Configure Distribution Services (CFGDSTSRV) command or the Add Distribution Queue (ADDDSTQ) command.

This is a required parameter.

Priority (PTY)

Specifies whether the normal priority or high priority portion of the specified queue is released.

The possible values are:
*NORMAL
  The normal priority queue is for those distributions with a service level of data low.

*HIGH
  The high priority queue is for those distributions with a service level of fast, status, or data high.
  Note: This value is not valid for a SystemView distribution services (SVDS) type of distribution queue.

This is a required parameter.

Examples

Example 1: Releasing the Normal Priority Portion of the Queue
RLSDSTQ DSTQ(CHICAGO) PTY(*NORMAL)

This command releases the normal priority portion of the CHICAGO distribution queue.

Example 2: Releasing the High Priority Portion of the Queue
RLSDSTQ DSTQ(ATLANTA) PTY(*HIGH)

This command releases the high priority portion of the ATLANTA distribution queue.

Error messages

*ESCAPE Messages

CPF8802
  Distribution queue &1 was not found.

CPF8805
  Special value for System name/Group not permitted or not used correctly.

CPF8806
  Value &1 not valid for system name or system group.

CPF881C
  High priority queue not allowed for *SVDS distribution queue &1

CPF8812
  Error occurred while processing distribution queues.

CPF8816
  QSNADS communications subsystem is not active.

CPF8817
  Distribution queue is held.

CPF9845
  Error occurred while opening file &1.

CPF9846
  Error while processing file &1 in library &2.

CPF9847
  Error occurred while closing file &1 in library &2.
Release File System Locks (RLSIFSLCK)

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

The Release File System Locks (RLSIFSLCK) command can be used to release all Network File System (NFS) byte-range locks held by a specified NFS client, or to release all byte-range locks (of any type) held on a specified object. This command should only be used to free resources that cannot be freed using normal means.

For more information about byte range locks, see the fcntl API in System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter or Network File System book, SC41-5714

Restrictions:
1. The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.
2. The user must have execute (*X) authority to the directories in the path name prefixes.
3. The user must have read (*R) authority to the object whose locks are being released.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMTLOCNAME</td>
<td>Remote location</td>
<td>Character value</td>
<td>Optional, Positional 1</td>
</tr>
<tr>
<td>OBJ</td>
<td>Object</td>
<td>Path name</td>
<td>Optional, Positional 2</td>
</tr>
</tbody>
</table>

Remote location (RMTLOCNAME)

Specifies the host name or internet address of a remote system whose NFS-related locks on local files are to be released.

To be successful, the remote system name must be valid. The user can assign host names to an internet address with the Work with TCP/IP host table entries option on the Configure TCP/IP menu (CFGTCP) command. Also, a remote name server can be used to map remote system names to internet addresses. Use the Change remote name server option on the CFGTCP menu to specify a remote name server.

Host names must follow these conventions:
• The first character must be either A through Z or 0 through 9.
• The last character must be either A through Z or 0 through 9.
• Uppercase and lowercase characters are allowed, but no significance is attached to the case.
• Blanks ( ) are not allowed.
• The special characters, period (.) and minus (-), are allowed.
• Parts of the name separated by periods (.) cannot exceed 63 characters in length.
• Names must be from 1 to 255 characters in length.

'remote-location-name'
   Specifies the host name or internet address of a remote system whose NFS-related locks on local files are to be released.

Either the RMTLOCNAME or Object (OBJ) parameter must be specified, but not both.

Object (OBJ)
Specifies the path name of an object on which all byte-range locks are to be released. This will release all locks on that object, regardless of the type of lock or the type of process that is holding them.

'path-name'
   Specifies the path name of the local object for which all locks are to be released.

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/iseries/infocenter.

Either the Remote location (RMTLOCNAME) or OBJ parameter must be specified, but not both.

Examples
Example 1: Releasing Locks for a Remote System
RLSIFSLCK RMTLOCNAME('rainbow1')

This command releases the NFS-related locks held on local files by the system named rainbow1.

Example 2: Releasing Locks for a Local Object
RLSIFSLCK OBJ('/CustAccounts/May')

This command releases all byte-range locks held on the object /CustAccounts/May.

Error messages
*ESCAPE Messages
CPFA09C
   Not authorized to object. Object is &1.

CPFA0A9
   Object not found. Object is &1.

CPFA0B6
   TOOBJ parameter is not allowed with a pattern in OBJ for CPY.

CPFA1B8
   *IOSYSCFG authority required to use &1.
iSeries: i5/OS Commands Starting with MRGTCPHT (Merge TCP/IP Host Table)

IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPHT (Merge TCP/IP Host Table)
Release Job (RLSJOB)

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

The Release Job (RLSJOB) command makes a job eligible for processing either after that job is held from processing by the Hold Job (HLDJOB) command, or if the job was submitted to the system as a held job by the Batch Job (BCHJOB) command or Submit Job (SBMJOB) command. The job being released can be on the job queue, on an output queue, or active in a subsystem (competing for system resources) when it is held. Releasing a job causes all threads within the job to be released. Spooled output files being held because *YES is specified for the Hold spooled files (SPLFILE) parameter in the Hold Job (HLDJOB) command are also released. Only those spooled output files which are on output queues in the library name space of the thread issuing this command (RLSJOB) will be released. If the Spooled file action (SPLFACN) job attribute is *DETACH and the job is ended while the spooled files are held, the spooled files cannot be released using the RLSJOB command. To release spooled files after the job has been removed from the system, use the Release Spooled File (RLSSPLF) command.

Restrictions: The issuer of the command must be running under a user profile which is the same as the job user identity of the job being released, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority.

The job user identity is the name of the user profile by which a job is known to other jobs. It is described in more detail in the Work Management book.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB</td>
<td>Job name</td>
<td>Qualified job name</td>
<td>Required,</td>
</tr>
<tr>
<td></td>
<td>Qualified 1: Job name</td>
<td>Name</td>
<td>Positional 1</td>
</tr>
<tr>
<td></td>
<td>Qualified 2: User</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualified 3: Number</td>
<td>000000-999999</td>
<td></td>
</tr>
<tr>
<td>DUPJOBOPT</td>
<td>Duplicate job option</td>
<td>*SELECT, *MSG</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Job name (JOB)

Specifies the name of the job being released.

This is a required parameter.

Qualifier 1: Job name

name Specify the name of the job being released.

Qualifier 2: User

name Specify the user name that identifies the user profile under which the job is started.
Qualifier 3: Number

000000-999999
   Specify the system-assigned job number.

Note: If no user name or job number is specified, all jobs currently in the system are searched for the job name. If more than one occurrence of the specified name is found, a qualified job name must be provided either explicitly or through the selection display. Refer to the Duplicate job option (DUPJOBOPT) parameter for more information.

Duplicate job option (DUPJOBOPT)
Specifies the action taken when duplicate jobs are found by this command.

*SELECT
   The selection display is shown when duplicate jobs are found during an interactive session. Otherwise, a message is issued.

*MSG  A message is issued when duplicate jobs are found.

Examples

Example 1: Releasing a Job for Processing
RLSJOB   JOB(PAYROLL)
This command releases the job PAYROLL for processing. If the corresponding HLDJOB command had specified SPLFILE(*YES), any spooled files for job PAYROLL are also released.

Example 2: Releasing a Job for Processing
RLSJOB   JOB(DEPTXYZ/PAYROLL)
This command releases job name PAYROLL that was submitted by a user through the user profile DEPTXYZ and later held. The qualified form of the job name is used when jobs with duplicate names exist in the system.

Error messages

*ESCAPE Messages

CPF1317
   No response from subsystem for job &3/&2/&1.

CPF1321
   Job &1 user &2 job number &3 not found.

CPF1332
   End of duplicate job names.

CPF1340
   Job control function not performed.
CPF1341
   Reader or writer &3/&2/&1 not allowed as job name.

CPF1343
   Job &3/&2/&1 not valid job type for function.

CPF1344
   Not authorized to control job &3/&2/&1.

CPF1349
   Job &3/&2/&1 not released, the job is not held.

CPF1351
   Function check occurred in subsystem for job &3/&2/&1.

CPF1352
   Function not done. &3/&2/&1 in transition condition.
Release Job Queue (RLSJOBQ)

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

The Release Job Queue (RLSJOBQ) command releases, for additional processing, the jobs on the specified job queue that were previously held by a HLDJOBQ (Hold Job Queue) command. If the jobs were held by something other than a HLDJOBQ command, they are not released.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBQ</td>
<td>Job queue</td>
<td>Qualified object name</td>
<td>Required,</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Job queue</td>
<td>Name</td>
<td>Positional 1</td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
</tbody>
</table>

### Job queue (JOBQ)

Specifies the job queue to be released for further processing.

This is a required parameter.

**Qualifier 1: Job queue**

*name* Specify the name of the job queue which you want to release.

**Qualifier 2: Library**

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

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

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

### Examples

RLSJOBQ JOBQ(QBATCH)

Jobs on the job queue QBATCH that were held by a HLDJOBQ command become eligible for processing, including jobs that were placed on the queue while it was being held. Specific jobs that were held by the HLDJOB command or that were put on the job queue in the held state are not released.
Error messages

*ESCAPE Messages

CPF2207
Not authorized to use object &1 in library &3 type *&2.

CPF3307
Job queue &1 in &2 not found.

CPF3330
Necessary resource not available.

CPF3423
Job queue &1 in library &2 not released. Job queue not held.
Release Job Schedule Entry (RLSJOBSCDE)

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

The Release Job Schedule Entry (RLSJOBSCDE) command allows you to release an entry, all entries, or a set of entries in the job schedule. Each job schedule entry contains the information needed to automatically submit a job to be run once or at regularly scheduled intervals.

If you release a job schedule entry, the job is not submitted immediately, even if the date and time at which it was scheduled to be submitted passed while the entry was held. The job is submitted on any future dates for which it is scheduled to be submitted.

Restrictions:
1. To release entries, you must have job control (*JOBCTL) special authority; otherwise you can release 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.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB</td>
<td>Job name</td>
<td>Generic name, name, *ALL</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>ENTRYNBR</td>
<td>Entry number</td>
<td>000001-999999, *ONLY, *ALL</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Job name (JOB)

Specifies the name of the job schedule entry.

*ALL  All of the job schedule entries for which you have authority are released. If JOB(*ALL) is specified, ENTRYNBR(*ALL) must also be specified.

generic-name

Specify the generic name used to find job schedule entries. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, then all entries with names that begin with the generic name, and for which the user has authority, are released. 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 release.
Entry number (ENTRYNBR)

Specifies the number of the job schedule entry you want to release. 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 released and an error message is sent.

*ALL
   All entries with the specified job name are released.

000001-999999
   Specify the number of the job schedule entry you want to release.

Examples

Example 1: Releasing All Job Schedule Entries
RLSJOBSCDE JOB(*ALL) ENTRYNBR(*ALL)

This command releases all the job schedule entries.

Example 2: Releasing an Individual Job Schedule Entry
RLSJOBSCDE JOB(PAYROLL) ENTRYNBR(*ONLY)

This command releases entry PAYROLL in the job schedule.

Example 3: Releasing a Generic Job Schedule Entry
RLSJOBSCDE JOB(PAY*) ENTRYNBR(*ALL)

This command releases all entries in the job schedule with the prefix PAY in their names.

Error messages

*ESCAPE Messages

CPF1628
   Job schedule entry &3 number &4 not found.

CPF1629
   Not authorized to job schedule &1.

CPF1630
   Not authorized to job schedule entry &3 number &4.

CPF1632
   Job schedule entry &3 number &4 damaged.

CPF1636
   More than one entry with specified entry job name found.

CPF1637
   Job schedule &1 in library &2 in use.
CPF1638
Job schedule entry &3 number &4 in use.

CPF1640
Job schedule &1 in library &2 does not exist.

CPF1641
Job schedule &1 in library &2 damaged.

CPF1645
No job schedule entries found for specified name.

CPF1646
Entry number must be *ALL when generic name specified.

CPF1648
&3 entries successfully released. &4 entries not released.

CPF1649
Entry number must be *ALL.
Release Output Queue (RLSOUTQ)

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

The Release Output Queue (RLSOUTQ) command releases the spooled files on the specified output queue. If the files were held by a Hold Spooled File (HLDSPLF) command or were created in a held state, they are not released.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTQ</td>
<td>Output queue</td>
<td>Qualified object name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Output queue</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
</tbody>
</table>

Output queue (OUTQ)

Specifies the output queue to be released so that the files on this queue can now be processed.

This is a required parameter.

Qualifier 1: Output queue

ame Specify the name of the output queue to be released.

Qualifier 2: Library

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

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

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

Examples

RLSOUTQ OUTQ(PRINTER)

On the output queue named PRINTER, spooled files that were held by a HLDSPLF command are released for further processing. This includes spooled files placed on the queue while it was being held, except for specific files that have been held by the HLDSPLF command or were put on the queue in hold.
Error messages

*ESCAPE Messages

CPF2207
Not authorized to use object &1 in library &3 type *&2.

CPF3330
Necessary resource not available.

CPF3357
Output queue &1 in library &2 not found.

CPF3424
Output queue &1 in library &2 not released. Output queue not held.
Release Reader (RLSRDR)

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

The Release Reader (RLSRDR) command releases the specified spooling reader so that it can continue to process jobs for the job queue. The specified reader was held by a previous HLDRDR (Hold Reader) command. Data was not lost when this reader was held.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDR</td>
<td>Reader</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
</tbody>
</table>

Reader (RDR)

Specifies the spooling reader to be released.

This is a required parameter.

name Specify the name of the reader to be released.

Examples

RLSRDR RDR(DISKETTE)

This command releases the diskette reader named DISKETTE for additional processing.

Error messages

*ESCAPE Messages

CPF1317
No response from subsystem for job &3/&2/&1.

CPF1340
Job control function not performed.

CPF1351
Function check occurred in subsystem for job &3/&2/&1.

CPF1352
Function not done. &3/&2/&1 in transition condition.
CPF3312
  Reader &1 neither active nor on job queue.

CPF3315
  Reader &3/&2/&1 not released. Reader not held.

CPF3330
  Necessary resource not available.

CPF3490
  Not authorized to specified reader.
Release Remote Phase (RLSRMTPHS)

Where allowed to run: All environments (*ALL)

Threadsafe: No

The Release Remote Phase (RLSRMTPHS) command initiates a session between the iSeries and a System/370 NetView Distribution Manager (NDM) host system. After the phase is released by NDM (or an unsuccessful attempt is made to do so), the session is ended.

The following considerations apply when running this command:

• The NDM plan specified by the PLAN parameter must exist and have been previously submitted to the NDM host application specified by the APPID parameter.
• The NDM phase specified by the PHASE parameter must exist and be part of the NDM plan specified by the PLAN parameter.
• The NDM phase specified by the PHASE parameter must be in a HELD state on the host system.
• The device specified by the DEV parameter must be a SNUF device and must be program start request (PSR) capable.
• This command runs only on a node which is currently functioning as a host interface node to the NDM host system. However, it is not restricted to releasing only those NDM phases whose destination is the node issuing the command. Any phase may be released for any node that shares the host interface node.

Restriction: To use this command, the user must be signed on as QPGMR or QSYSOPR, or have *ALLOBJ authority.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHASE</td>
<td>Phase</td>
<td>Communications name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>PLAN</td>
<td>Plan</td>
<td>Communications name</td>
<td>Required, Positional 2</td>
</tr>
<tr>
<td>APPID</td>
<td>Application identifier</td>
<td>Name</td>
<td>Required, Positional 3</td>
</tr>
<tr>
<td>RMTLOCNAME</td>
<td>Remote location</td>
<td>Communications name</td>
<td>Required, Positional 4</td>
</tr>
<tr>
<td>DEV</td>
<td>Device</td>
<td>Name</td>
<td>Required, Positional 5</td>
</tr>
</tbody>
</table>
Phase (PHASE)
Specifies the name of the NetView Distribution Manager phase that is released. This phase must exist on the NDM host system as part of the plan specified by the PLAN parameter, and must be in a HELD state.

This is a required parameter.

Plan (PLAN)
Specifies the name of the NetView Distribution Manager plan that contains the phase that is released. This plan must exist on the NDM host.

This is a required parameter.

Application identifier (APPID)
Specifies the name of the NetView Distribution Manager application under which the phase name specified by the PHASE parameter was submitted. This is the same name by which NDM was made known to MVS when it was generated.

This is a required parameter.

Release Remote Phase (RMTLOCNAME)
Specifies the name of the remote location with which this device communicates. This should be the same name that is entered in the RMTLOCNAME parameter of the device specified by the DEV parameter.

This is a required parameter.

Device description (DEV)
Specifies the device name of the iSeries device to be used for the communications session started as a result of this command. The device must be a SNUF device and must be PSR capable.

This is a required parameter.

Examples
RLSRMTPHS PHASE(MESSAGE) PLAN(ALEXPLAN) APPID(DSXNDM)
RMTLOCNAME(A083187) DEV(SNUFDEV)

This command initiates a session using device SNUFDEV with remote location name A083187 to connect with the System/370 NetView Distribution Manager host application DSXNDM. After the session
connection is made, phase MESSAGE, as part of plan ALEXPLAN, attempts to release. If the release is successful, message CPC8889 (Phase MESSAGE released by NetView Distribution Manager) is sent. If the release is not successful, message CPF8880 (Phase MESSAGE not released by Netview Distribution Manager) is sent.

---

**Error messages**

*ESCAPE Messages*

CPF8880

Phase &1 not released by NetView Distribution Manager.
Release Spooled File (RLSSPLF)

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

The Release Spooled File (RLSSPLF) command releases the specified file on an output queue. The file being released is always produced from the beginning of the file. The RLSSPLF command can release a spooled file that was held by:

- A HLDSPLF command
- HOLD(*YES) being specified in the device file or on an override command
- SAVE(*YES) being specified in the device file, on an override command, or in the CHGSPLFA command
- SAVE(*IMMED) being specified in the CHGSPLFA command
- A HLDWTR command and a RLSWTR command with OPTION(*BYPASS) specified
- The operator canceling a system request to put forms on the printer

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>Spooled file</td>
<td>Name, *SELECT</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>JOB</td>
<td>Job name</td>
<td>Single values: * Qualified job name</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Job name</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: User</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 3: Number</td>
<td>000000-999999</td>
<td></td>
</tr>
<tr>
<td>SPLNBR</td>
<td>Spooled file number</td>
<td>1-999999, *ONLY, *LAST, *ANY</td>
<td>Optional, Positional 3</td>
</tr>
<tr>
<td>JOBSYSNAME</td>
<td>Job system name</td>
<td>Name, *ONLY, *CURRENT, *ANY</td>
<td>Optional</td>
</tr>
<tr>
<td>CRDATE</td>
<td>Spooled file created</td>
<td>Single values: *ONLY, *LAST Other values: Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Creation date</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Creation time</td>
<td>Time, *ONLY, *LAST</td>
<td></td>
</tr>
<tr>
<td>SELECT</td>
<td>Select files for</td>
<td>Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: User</td>
<td>Name, *CURRENT, *ALL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Print device</td>
<td>Name, *ALL, *OUTQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 3: Form type</td>
<td>Character value, *ALL, *STD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 4: User data</td>
<td>Character value, *ALL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 5: ASP</td>
<td>1-32, *ALL, *ASPDEV</td>
<td></td>
</tr>
<tr>
<td>ASPDEV</td>
<td>ASP device</td>
<td>Name, *, *SYSBAS, *CURASPGRP</td>
<td>Optional</td>
</tr>
</tbody>
</table>
**Spooled file (FILE)**

Specifies the name of the spooled file that is to be released so that it can now be written to an output device such as a printer or diskette.

This is a required parameter.

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

name  Specify the name of the spooled file to release.

**Job name (JOB)**

Specifies the job that created the spooled file being released for additional processing.

Single values

*  The job that issued this Release Spooled File (RLSSPLF) command is the same job that produced the spooled file.

Qualifier 1: Job name

name  Specify the name of the job that created the file you want to release.

Qualifier 2: User

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

Qualifier 3: Number

000000-999999  Specify the system-assigned job number of the job that created the spooled file to be released.

**Spooled file number (SPLNBR)**

Specifies the number of the spooled file being released.

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

*LAST  
The highest-numbered spooled file with the specified file name is the file to release.

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

1-999999  Specify the number of the spooled file to release that has the specified file name.
Job system name (JOBSYSNAME)

Specifies the name of the system where the job that created the spooled file (JOB parameter) ran. This parameter is considered after the job name, user name, job number, spooled file name, and spooled file number parameter requirements have been met.

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

*CURRENT
The spooled file created on the current system with the specified job name, user name, job number, spooled file name, spooled file number, and create date and time is used.

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

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

Spooled file created (CRTDATE)

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

Single values

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

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

Element 1: Creation date

date Specify the date the spooled file was created.

Element 2: Creation time

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

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

time Specify the time the spooled file was created.
Select files for (SELECT)

Specifies which group of files should be selected to be released. Positional values can be specified to select the files: the user that created the file, the device that the file is queued for, the form type specified, the user data tag associated with the file, or the auxiliary storage pool the file is in. Only files that meet each of the values are selected.

Element 1: User

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

*ALL    Files created by all users are selected.

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

Element 2: Print device

*ALL    Files on any device-created or user-created output queue are selected.

*OUTQ   All files on any user-created output queue are selected. A user-created output queue is any output queue that is not automatically created by a device. A user-created output queue does not generally have the same name as a device, but if it does, it does not reside in library QUSRsys.

name    Specify a device name. Only files on the device created output queue for that device are selected. A device created output queue is one that has the same name as a device and resides in the QUSRsys library. Unless it already exists, it will automatically be created by the system when the device is created. A device created output queue cannot be deleted.

Element 3: Form type

*ALL    Files for all form types are selected.

*STD    Only files that specify the standard form type are selected.

form-type    Specify the form type to select the file.

Element 4: User data

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

user-data    Specify the user data tag to select the file.

Element 5: ASP

*ALL    All files as specified in the Auxiliary Storage Pool Device (ASPDEV) parameter are selected.

*ASPDEV  Files specified in the Auxiliary Storage Pool Device (ASPDEV) parameter are selected.

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

ASP device (ASPDEV)

Specifies the auxiliary storage pool device name from which spooled files are to be selected. This parameter is only valid if the ASP number (ASP) element of the Select parameter is *ALL or *ASPDEV.

Files which are found in the ASPs that are currently part of the thread’s library name space are
selected. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and if the thread has an ASP group, the primary and secondary ASPs in the thread’s ASP group.

*SYSBAS
Files which are found in the system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) are selected.

*CURASPGRP
Files which are found in the primary and secondary ASPs in the thread’s ASP group are selected. If no ASP group is associated with the thread, an error will be issued.

name Specify the name of the auxiliary storage pool device description. Files which are found in the specified primary or secondary ASP are selected. Only primary or secondary ASPs which are in the thread’s ASP group may be specified. If no ASP group is associated with the thread, an error will be issued.

Examples
RLSSPLF   FILE(STOCK14)  JOB(000047/SMITH/Master)

This command releases the spooled file named STOCK14 created in the job named MASTER. The file can now be selected for processing by the spooling writer. The job was run under the user profile named SMITH and was assigned the job number 000047 by the system.

Error messages
*ESCAPE Messages
CPF337E
ASP device &1 not in current ASP group for thread.

CPF337F
ASP device &1 not allowed with ASP number &2.

CPF33D0
Printer &1 does not exist.

CPF33D1
User &1 does not exist.

CPF3303
File &1 not found in job &5/&4/&3.

CPF3304
File &1 number &8 cannot be released.

CPF3309
No files named &1 are active.

CPF3322
File &1 number &8 not released.

CPF3330
Necessary resource not available.

CPF3340
More than one file with specified name found in job &5/&4/&3.
CPF3342
Job &5/&4/&3 not found.

CPF3343
Duplicate job names found.

CPF3344
File &1 number &8 no longer in the system.

CPF3357
Output queue &1 in library &2 not found.

CPF3362
Objects in QTEMP not valid for parameter values.

CPF3492
Not authorized to spooled file.

CPF9825
Not authorized to device &1.

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

CPF8BED
Device description &1 not correct for operation.
Release Writer (RLSWTR)

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

The Release Writer (RLSWTR) command releases a held writer so that files on the output queue can now be processed. If the writer was writing a file when it was held, the writer can be released to resume writing this same file or it can be released to start writing the next file. In any case, data from the file that was being written when the Hold Writer (HLDWTR) command was issued is not lost.

Note: You must specify a value for either the Resume writing at (OPTION) parameter or the Starting page (PAGE) parameter. You cannot specify both.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>WTR</td>
<td>Writer</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>OPTION</td>
<td>Resume writing at</td>
<td>-32766-32766, *CURRENT, *BEGIN, *BYPASS</td>
<td>Optional</td>
</tr>
<tr>
<td>PAGE</td>
<td>Starting page</td>
<td>Integer</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Writer (WTR)

Specifies the spooling writer to release. After releasing the writer, you will be able to process the files on the output queue (provided the output queue is not held).

This is a required parameter.

name Specify the name of the writer to be released.

Resume writing at (OPTION)

Specifies the point in the file where you want the writer to begin. Only the first option *CURRENT can be specified when the writer is not producing a file. The last three options (*BYPASS, +number, and -number) can be specified only if the writer was held while it was producing this file. Also, the only valid option for the diskette writer is *CURRENT.

*CURRENT
The writer is released at the point where it had been held by the Hold Writer (HLDWTR) command.

*BEGIN
The writer is released at the beginning of the current file.
*BYPASS
  The writer is released at the beginning of the next file. The current file is to be implicitly held on the queue.

+number
  The writer is released n number of pages past the point where it was held.

-number
  The writer is to be released n number of pages before the point where it was held.

Starting page (PAGE)
Specifications the page where the writer starts printing. This parameter is mutually exclusive with the Resume writing at (OPTION) parameter and is only valid for a printer writer. This parameter can be specified only if the writer was held while producing the file.

integer
  Specify the page number in this file where you want the writer to start printing.

Examples

Example 1: Releasing a Writer at Beginning of File
RLSWTR  WTR(PRINTER)  OPTION(+BEGIN)
This command releases writer PRINTER to begin producing the current file at its beginning.

Example 2: Releasing Writer at Specified Point
RLSWTR  WTR(PRTR)  OPTION(-3)
This command releases writer PRTR to begin printing again at a point three pages before the point where the writer was held. That is, the last three pages previously printed are the first three pages printed this time.

Example 3: Starting Printing on Page Ten
RLSWTR  WTR(PRTR)  PAGE(10)
This command releases writer PRTR to start printing again at page ten.

Error messages

*ESCAPE Messages

CPF1317
  No response from subsystem for job &3/&2/&1.

CPF1340
  Job control function not performed.

CPF1352
  Function not done. &3/&2/&1 in transition condition.
CPF3313
Writer &1 not active nor on job queue.

CPF3314
PAGE parameter allowed only for print writers.

CPF3316
Writer &3/&2/&1 not released because writer not held.

CPF3317
OPTION parameter value not allowed for diskette writer.

CPF3330
Necessary resource not available.

CPF3331
Not authorized to control writer &3/&2/&1.

CPF3334
Previous hold to writer &3/&2/&1 pending.
Remove Directory (RMDIR)

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

The Remove Directory (RMDIR) 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 is an alias for the Remove Directory (RMVDIR) command and can also be issued using the following alternative command names:
- RD
- RMVDIR

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.
Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR</td>
<td>Directory</td>
<td>Path name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>RMVLNK</td>
<td>Remove link</td>
<td>*NO, *YES</td>
<td>Optional</td>
</tr>
</tbody>
</table>

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/iseries/infocenter.

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.

Examples

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

**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.
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

CPFA0B2
No objects satisfy request.

CPFA0B7
&1 directories removed. &2 directories not removed.
Remove Access Code (RMVACC)

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

The Remove Access Code (RMVACC) command allows you to remove from the system an access code previously defined by the Add Access Code (ADDACC) command.

NOTES:
1. This command can take a long time to run because it must update each object in the document library that has been assigned the access code being removed.
2. This command removes the access code from all filed documents, from all users authorized to the access code, and from the system.

Restrictions:
To use this command, you must have all object (*ALLOBJ) special authority.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>Document access code</td>
<td>Values (up to 300 repetitions): 1-2047</td>
<td>Required, Positional 1</td>
</tr>
</tbody>
</table>

Document access code (ACC)

Specifies the access code or access codes being removed from the system. If the access code specified is not defined on the system, a diagnostic message is sent, and any additional access codes specified are processed.

1-2047 Specify an access code to be removed from the system.

This is a required parameter.

Examples

RMVACC ACC(300)

This command removes access code 300 from the system.
Error messages

*ESCAPE Messages

CPF90A5
   Access codes could not be removed.

CPF9009
   System requires file &1 in &2 be journaled.

CPF9011
   &1 access codes removed, &2 not removed.

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.
Remove Autostart Job Entry (RMVAJE)

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

The Remove Autostart Job Entry (RMVAJE) command removes an autostart job entry from the specified 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 that subsystem description.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBSD</td>
<td>Subsystem description</td>
<td>Qualified object name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Subsystem</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>description</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>JOB</td>
<td>Job name</td>
<td>Name</td>
<td>Required, Positional 2</td>
</tr>
</tbody>
</table>

Subsystem description (SBSD)

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

This is a required parameter.

Qualifier 1: Subsystem description

*name  Specify the name of the subsystem description from which the autostart job entry is being 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

---

Job name (JOB)

Specifies the simple name of the job that is started from the autostart job entry.

This is a required parameter.

name  Specify the simple name of the job that is started from the autostart job entry.

---

Examples

RMVAJE  SBSD(MYLIB/PAYROLL)  JOB(INITIAL)

This command removes job entry named INITIAL that starts automatically from the PAYROLL subsystem description in the library MYLIB.

---

Error messages

*ESCAPE Messages

CPF1619
   Subsystem description &1 in library &2 damaged.

CPF1697
   Subsystem description &1 not changed.
The Remove Alert Description (RMVALRD) command allows you to remove an alert description that was added previously by the ADDALRD command. More information on alerts is in the Alerts Support book, SC41-5413.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSGID</td>
<td>Message identifier</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>ALRTBL</td>
<td>Alert table</td>
<td>Qualified object name</td>
<td>Required, Positional 2</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Alert</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>table</td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB</td>
</tr>
</tbody>
</table>

#### Message identifier (MSGID)

Specifies the message ID for the alert description that is removed.

This is a required parameter.

**name** Specify the message identifier.

#### Alert table (ALRTBL)

Specifies the alert table from which this alert description is removed.

This is a required parameter.

**Qualifier 1: Alert table**

**name** Specify the name of the alert table that is used.

**Qualifier 2: Library**

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

*CURLIB* The current library is searched for the alert 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 alert table is located.
Examples

RMVALRD  MSGID(USR1234)  ALRTBL(USER/USRMGS)

This command removes the alert description for message identifier USR1234.

Error messages

*ESCAPE Messages

CPF1A04
Alert code &1 not found in alert table &2.

CPF1A05
Alert table &1 in &2 damaged.

CPF2499
Message identifier &1 not allowed.

CPF7BB1
Alert description not found.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9811
Program &1 in library &2 not found.

CPF9812
File &1 in library &2 not found.

CPF9814
Device &1 not found.

CPF9820
Not authorized to use library &1.

CPF9821
Not authorized to program &1 in library &2.

CPF9822
Not authorized to file &1 in library &2.
CPF9825
   Not authorized to device &1.

CPF9830
   Cannot assign library &1.

CPF9831
   Cannot assign device &1.
Remove Auth List Entry (RMVAUTLE)

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

The Remove Authorization List Entry (RMVAUTLE) command removes user entries from an authorization list. The authorization list must already exist.

Restrictions:
- Only the owner of the authorization list, a user with authorization list management authority (*AUTLMGT) on the authorization list, or who has all object (*ALLOBJ) special authority can use this command.
- The user with *AUTLMGT authority can only remove a user if the user with *AUTLMGT authority has at least the same specific authorities as the user to be removed.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTL</td>
<td>Authorization list</td>
<td>Generic name, name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>USER</td>
<td>User</td>
<td>Values (up to 50 repetitions): Name</td>
<td>Required, Positional 2</td>
</tr>
</tbody>
</table>

Authorization list (AUTL)

Specifies the authorization list from which the user names are to be removed. The authorization list must exist when the RMVAUTLE command is run.

This is a required parameter.

generic-name

Specify the generic name of the authorization lists to be changed.

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

name

Specify the name of the authorization list to be changed.
**User (USER)**

Specifies one or more user names to be removed from the authorization list. A maximum of 50 user names can be specified.

This is a required parameter.

**Examples**

RMVAUTLE AUTL(PAYROLL) USER(TOM JULIE KAREN)

This command removes users TOM, JULIE, and KAREN from the authorization list PAYROLL.

**Error messages**

*ESCAPE Messages*

CPF2253
   No objects found for &1 in library &2.

CPF2281
   The users specified do not exist on the system.

CPF2283
   Authorization list &1 does not exist.

CPF2284
   Not authorized to change authorization list &1.

CPF2285
   &1 errors removing users, &2 authorization lists processed.

CPF2288
   *PUBLIC cannot be removed from an authorization list.

CPF2289
   Unable to allocate authorization list &1.
Remove Breakpoint (RMVBKP)

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

The Remove Breakpoint (RMVBKP) command removes one or more breakpoints from the specified program being debugged. It can also remove all breakpoints from all programs in debug mode.

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 breakpoints from a bound program.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
</table>
| STMT    | Statement identifier | Single values: *, *ALL  
Other values (up to 10 repetitions): Character value | Optional, Positional 1 |
| PGM     | Program       | Name, *DFTPGM, *ALL                         | Optional, Positional 2 |

Statement identifier (STMT)

Specifies which HLL (high-level language) statements or machine instructions in a program have their breakpoints removed. Breakpoints can be removed from a specified program, the program specified in the Program (PGM) parameter or from the most recent program that has reached a breakpoint, i.e., a value of * specified for the STMT parameter. If a program is specified, one or more statement identifiers can be specified or all the breakpoints can be specified. If a value of * is specified for the STMT parameter, the breakpoint that the most recently stopped program has reached is removed. Also, all breakpoints can be removed from all programs in debug mode.

Single values
*  The most recent breakpoint at which a program is currently stopped is the breakpoint removed.
*ALL  All breakpoints in the specified program are removed.

Other values (up to 10 repetitions)
character-value  Specify the statement identifier to be removed for the program specified by the PGM parameter. No more than 10 identifiers can be specified.
**Program (PGM)**

Specifies the program from which the specified breakpoints are removed. This parameter can be specified only if a value of * has not been specified for the **Statement identifier (STMT)** parameter.

* **DFTPGM**
  
  The default program is the program whose breakpoints are removed.

* **ALL**
  
  All programs currently in debug mode have their breakpoints removed. This value can be specified only if no value for the STMT parameter has been supplied.

**name**

Specify the name of the program from which the specified breakpoints are removed.

---

**Examples**

**RMVBKP STMT(100)**

This command removes the breakpoint that is on statement 100 from the default program.

---

**Error messages**

* **ESCAPE Messages**

**CPF1999**

Errors occurred on command.
Remove Binding Directory Entry (RMVBNDDIRE)

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

The Remove Binding Directory Entry (RMVBNDDIRE) command removes an entry from the binding directory.

Restrictions:
- You must have read (*READ) and object operational (*OBJOPR) authority for the library where the binding directory is being updated.
- You must have *OBJOPR and delete (*DLT) authority to the binding directory.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BNDDIR</td>
<td>Binding directory</td>
<td>Qualified object name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Binding directory</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL, *CURLIB, *USRLIBL</td>
<td></td>
</tr>
<tr>
<td>OBJ</td>
<td>Object specifications</td>
<td>Values (up to 50 repetitions): Element list</td>
<td>Optional, Positional 2</td>
</tr>
<tr>
<td></td>
<td>Element 1: Object</td>
<td>Qualified object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Object</td>
<td>Generic name, name, *ALL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIBL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Object type</td>
<td>*SRVPGM, *MODULE</td>
<td></td>
</tr>
</tbody>
</table>

Binding directory (BNDDIR)

Specifies the binding directory from which an entry is removed.

This is a required parameter.

Qualifier 1: Binding directory

name Specify the name of the binding directory to be updated.

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.

*USRLIBL Only the libraries in the user portion of the job’s library list are searched.
**name** Specify the name of the library to be searched.

---

**Object specifications (OBJ)**

Specifies one or more object names to be removed from the binding directory.

You can specify 50 values for this parameter.

**Element 1: Object**

**Qualifier 1: Object**

*ALL All objects with the specified type are removed from the specified library.

**generic-name**

Specify the generic name of the objects 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. If the complete object name is specified, and multiple libraries are searched, multiple objects can be removed, only if *ALL or *ALLUSR library values can be specified for the name.

**name** Specify the name of the object to remove.

**Qualifier 2: Library**

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

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

**Element 2: Object type**

**SRVPGM**

Indicates the object being removed is a service program.

**MODULE**

Indicates the object being removed is a module.

---

**Examples**

RMVBNDDIR

BNDDIR(SOURCE) OBJ(LIST)

This command allows you to remove the object LIST from the binding directory SOURCE.
**Error messages**

*ESCAPE Messages*

CPF5D01
    Binding directory &1 in library &2 is not usable.

CPF5D09
    Object &2/&1 type &3 was not found in binding directory &4 in library &5.

CPF980F
    Binding directory &1 in library &2 not found.

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.
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPPHT (Merge TCP/IP Host Table)
Remove Cfg List Entries (RMVCFGLE)

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

The Remove Configuration List Entries (RMVCFGLE) command removes entries from a configuration list.

Note: The user may also use the full screen entry display of the Change Configuration List (CHGCFGLE) command to add, remove, or change entries in an existing list except for the configuration list TYPE(*SNAPASTHR).

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPNLCL</td>
<td>APPN local location entry</td>
<td>Values (up to 50 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Local location name</td>
<td>Communications name</td>
<td></td>
</tr>
<tr>
<td>APPNRMT</td>
<td>APPN remote location entry</td>
<td>Values (up to 50 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Remote location name</td>
<td>Generic name, name, *ANY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Remote network identifier</td>
<td>Communications name, *NETATR, *NONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 3: Local location name</td>
<td>Communications name, *NETATR</td>
<td></td>
</tr>
<tr>
<td>CFGL</td>
<td>Configuration list name</td>
<td>Name</td>
<td>Optional</td>
</tr>
<tr>
<td>ASYNCADR</td>
<td>Async network address entry</td>
<td>Values (up to 50 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Network address</td>
<td>Character value</td>
<td></td>
</tr>
<tr>
<td>ASYNCLOC</td>
<td>Async remote location entry</td>
<td>Values (up to 50 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Remote location name</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>RTLPASTHR</td>
<td>Retail pass-through entry</td>
<td>Values (up to 50 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: Retail device</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>FTRCPNAME</td>
<td>Filtered control point name</td>
<td>Generic name, name, *ANY</td>
<td>Optional</td>
</tr>
<tr>
<td>FTRCPNETID</td>
<td>Filtered CP network identifier</td>
<td>Communications name, *NETATR</td>
<td>Optional</td>
</tr>
<tr>
<td>LCLLOCNAME</td>
<td>Local location name</td>
<td>Generic name, name, *ANY</td>
<td>Optional</td>
</tr>
<tr>
<td>SNAPASTHR</td>
<td>SNA pass-through entry</td>
<td>Values (up to 254 repetitions): Element list</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Element 1: SNA pass-through group name</td>
<td>Communications name</td>
<td></td>
</tr>
</tbody>
</table>
Configuration list type (TYPE)

Specifies one of the possible configuration list types.

This is a required parameter.

*APPNDIR
   An advanced peer-to-peer networking (APPN) directory search filter configuration list is used. The user can specify one APPN directory search filter entry in the configuration list.

*APPNLCL
   An APPN local location list is used. The user can specify up to 476 APPN local location entries in the configuration list.

*APPNRMT
   An APPN remote location list is used. The user can specify up to 1898 APPN remote location entries in the configuration list.

*APPNSSN
   An APPN session end point filter configuration list is used. The user can specify one APPN session entry in the configuration list.

*ASYNCADR
   An asynchronous network address list is used. The user can specify up to 294 asynchronous network address entries in the configuration list.

*ASYNCLOC
   An asynchronous remote location list is used. The user can specify up to 32000 asynchronous remote location entries in the configuration list.

*RTLPASTHR
   A retail pass-through list is used. The user can key up to 450 retail pass-through entries in the configuration list.

*SNAPASTHR
   An SNA pass-through list is used. The user can key one SNA pass-through entry in the configuration list.

APPN local location entry (APPNLCLE)

Specifies the APPN local location entry. This value is required if *APPNLCL is specified for the Configuration list type prompt (TYPE parameter).

You can enter multiple values for this parameter.

A maximum of 50 entries can be specified directly for this parameter. An entry consists of a value from each of the following elements.

local-location-name
   Specify the local location of the entry being removed from the configuration list.
APPN remote location entry (APPNRMTE)

Specifies the APPN remote location entry. This value is required if *APPNRMT is specified for the Configuration list type prompt (TYPE parameter).

You can enter multiple values for this parameter.

A maximum of 50 entries can be specified directly for this parameter. An entry consists of a value from each of the following elements.

- **remote-location-name**
  Specify the remote location of the entry being removed from the configuration list.

- **remote-network-identifier**
  Specify the remote network identifier of the entry being removed from the configuration list.

- **local-location-name**
  Specify the local location of the entry being removed from the configuration list.

Configuration list (CFGL)

Specifies the name of the configuration list. This value is required and valid only when the configuration list is an asynchronous network address list (*ASYNCADR is specified for the Configuration list type (TYPE) parameter). The list types have system-supplied names: QAPPNLCL, QAPPNRMT, QASYNCADR, QASYNCCLOC, QRTLAPASTHR, and QSNAPASSTHR.

This is a required parameter.

Async network address entry (ASYNCADRE)

Specifies the asynchronous network address entry. This value is required if *ASYNCADR is specified for the Configuration list type prompt (TYPE parameter).

You can enter multiple values for this parameter.

A maximum of 50 entries can be specified directly for this parameter. An entry consists of a value from each of the following elements.

- **network-address**
  Specify the network address of the entry being removed from the configuration list.
  
  **Note:** All entries having the same network address as the one you specify are removed from the configuration list.

Async remote location entry (ASYNCLOCE)

Specifies the asynchronous remote location entry. This value is required if *ASYNCLOC is specified for the Configuration list type prompt (TYPE parameter).

You can enter multiple values for this parameter.
A maximum of 50 entries can be specified directly for this parameter. An entry consists of a value from each of the following elements.

*remote-location-name*

Specify the remote location of the entry being removed from the configuration list.

---

**Retail pass-through entry (RTLPASTHRE)**

Specifies the retail pass-through entry. This value is required if *RTLPASTHR* is specified for the Configuration list type prompt (TYPE parameter).

You can enter multiple values for this parameter.

A maximum of 50 entries can be specified directly for this parameter. An entry consists of a value from each of the following elements.

*Retail-device-name*

Specify the retail device name of the entry being removed from the configuration list.

---

**Filtered control point name (FTRCPNAME)**

Specifies the control point name of the adjacent control point that is being filtered by the local system when a directory search request is made.

**Note:** This parameter is valid only if TYPE(*APPNDIR) is specified.

*ANY* Any control point name is filtered.

*generic*-filtered-CP-name

Specify the generic control point name (part of a name followed by an asterisk) of the adjacent control point(s) being filtered. The generic control point name allows one directory entry to be defined for all control points, in a single network, with a name that matches the characters preceding an asterisk (*).

*filtered-CP-name*

Specify the control point name of the adjacent control point being filtered.

---

**Filtered CP network identifier (FTRCPNETID)**

Specifies the control point network identifier of the adjacent control point being filtered by the local system when a directory search request is made.

**Note:** This parameter is valid only if TYPE(*APPNDIR) is specified.

*NETATR*

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

*filtered CP-network-ID*

Specify the control point network identifier of the adjacent control point being filtered by the local system.
Local location name (LCLLOCNAME)

Specifies the local location name being supplied by the caller that is being filtered by the local system. When the local system is initiating a session, this is the local location name being used. When a bind is received from another system, this is the Secondary Logical Unit (SLU) name being used.

**Note:** This parameter is valid only if TYPE(*APPNSSN) is specified.

*ANY  Any local location name will be filtered by the local system.

generic*-local-location-name

Specify the generic local location name (part of a name followed by an asterisk) of the local location(s) being filtered. The generic local location name allows one entry to be defined for all local location names, on the system, with a name that matches the characters preceding an *.

local-location-name

Specify the local location name that is being filtered by the local system.

SNA pass-through entry (SNAPASTHRE)

Specifies the SNA pass-through entry. This value is required if TYPE(*SNAPASTHR) is specified. One group entry can be specified for this parameter.

Examples

RMVCFGLE  TYPE(*ASYNCLOC)  ASYNCLOCE(RMTLOC1)

This command removes the configuration list entry RMTLOC1 from the asynchronous remote location list QASYNCLOC.

Error messages

*ESCAPE Messages

CPF260F  Configuration list &1 not found.

CPF261C  Index for configuration list &1 not changed.

CPF261D  Index for configuration list &1 not changed.

CPF2625  Not able to allocate object &1.

CPF263A  CFGL type &1 does not match existing type &2.

CPF2634  Not authorized to object &1.
CPF2663
Configuration list &1 previously deleted.

CPF2666
Cannot remove all entries from configuration list &1.
Remove Cluster Node Entry (RMVCLUNODE)

Where allowed to run: All environments (*ALL)

Threadsafe: No

The Remove Cluster Node Entry (RMVCLUNODE) command is used to remove a node from a cluster. The node specified will be removed from the cluster membership list and will no longer be considered a member of the cluster. The node will also be removed from the membership of the device domain to which it belongs. The cluster resource group objects on the node being removed are deleted only if the node has a status of Active or if this command is called on the node that is being removed.

A node can be removed regardless of its status. If this command is called on a node with a status of Active, any node in the cluster can be removed. If this command is called on a node with a status of Inactive or Failed, only the node running the command can be removed. To remove a node that is not active, this command should be called on the node being removed and on a node in the cluster that is active.

If the cluster is partitioned and any node in the partition is removed, then that node must be removed in all other partitions in order for an automerge to complete successfully.

If the node being removed is active, the cluster resource group exit program will be passed an action code of Remove Node. The exit program on all other nodes in the recovery domain of the cluster resource group will be passed an action code of Failover. If the node being removed is the primary node for a device cluster resource group, ownership of the hardware associated with the cluster resource group will be moved to a backup node. If there are no backup nodes or all the backup nodes are either inactive or in a different cluster partition, ownership of the hardware is left with the node being removed.

If the node being removed is inactive, the cluster resource group exit program will be passed an action code of Remove Node on all nodes in the recovery domain. Ownership of hardware associated with a device cluster resource group will not be changed but will remain with the node being removed. If the node being removed is inactive, the cluster resource group exit program will be called with an action code of Delete Command on the node being removed if the command is run on the node being removed.

If the node being removed is a member of a device domain and it later will be added back to a cluster, the node most likely will 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.

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. If all of the nodes in the cluster have a status of New, this command can only be called on the node where the cluster was originally created.
4. There must be more than one node in the membership list.
Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLUSTER</td>
<td>Cluster</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>NODE</td>
<td>Node identifier</td>
<td>Name</td>
<td>Required, Positional 2</td>
</tr>
</tbody>
</table>

Cluster (CLUSTER)

Specifies the cluster that contains the node to be removed.

This is a required parameter.

*name*  Specify the name of the cluster.

Node identifier (NODE)

Specifies the node identifier to be removed.

This is a required parameter.

*name*  Specify the name of the node to be removed.

Examples

RMVCLUNODE  CLUSTER(MYCLUSTER)  NODE(RCHCST01)

This command removes the node RCHCST01 from the cluster membership for cluster MYCLUSTER. Cluster Resource Services is ended on node RCHCST01.

Error messages

*ESCAPE Messages*

*CPF1999*

Errors occurred on command.
Remove Communications Entry (RMVCME)

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

The Remove Communications Entry (RMVCME) command removes a communications entry from an existing 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. All jobs that are active through the communications entry being removed must be ended before this command can be run.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBSD</td>
<td>Subsystem description</td>
<td>Qualified object name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td></td>
<td>Qualifier 1: Subsystem description</td>
<td>name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qualifier 2: Library</td>
<td>Name, *LIRL, *CURLIB</td>
<td></td>
</tr>
<tr>
<td>RMTLOCNAME</td>
<td>Remote location</td>
<td>Communications name</td>
<td>Optional</td>
</tr>
<tr>
<td>MODE</td>
<td>Mode</td>
<td>Name, *ANY</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Subsystem description (SBSD)

Specifies the name and library of the subsystem description from which the communications entry is being removed.

This is a required parameter.

Qualifier 1: Subsystem description

name Specify the name of the subsystem for the communications entry that is being removed.

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

Qualifier 2: Library

*LIRL 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 for the communications entry that is being removed.

---

**Device (DEV)**

Specifies the name of the device description, or the type of the device, for which the communications entry is being removed.

**Note:** You must specify a value on either this parameter or the **Remote location (RMTLOCNAME)** parameter but not for both.

*ALL The *ALL communications entry is removed.

*APPC The *APPC communications entry is removed.

*ASYNC The *ASYNC communications entry is removed. This value is valid only when *ANY is specified on the Mode (MODE) parameter.

*BSCEL The *BSCEL communications entry is removed. This value is valid only when *ANY is specified on the Mode (MODE) parameter.

*FINANCE The *FINANCE communications entry is removed. This value is valid only when *ANY is specified on the Mode (MODE) parameter.

*INTRA The *INTRA communications entry is removed. This value is valid only when *ANY is specified on the Mode (MODE) parameter.

*RETAIL The *RETAIL communications entry is removed. This value is valid only when *ANY is specified on the Mode (MODE) parameter.

*SNUF The *SNUF communications entry is removed. This value is valid only when *ANY is specified on the Mode (MODE) parameter.

generic-name Specify the name of the generic communications entry being removed.

name Specify the name of the device description for which the communications entry is being removed.

---

**Remote location (RMTLOCNAME)**

Specifies the name of the remote location for which the communications entry is removed.

**Note:** You must specify either this parameter or the **Device (DEV)** parameter, but not both.

*communications-name Specify the name of the remote location used with this communications entry.
Mode (MODE)

Specifies the name of the mode of the device specified on the Device (DEV) parameter or the remote location specified on the Remote location (RMTLOCNAME) parameter for which the communications entry is removed.

*ANY The communications device or remote location name with a mode name of *ANY is to be removed.

name Specify the name of the mode entry of the communications device or remote location name for which the communications entry is to be removed.

Examples

RMVCMNE  SBSD(LIB2/SBS1)  DEV(COMDEV)

This command removes the communications device entry for the device COMDEV from the subsystem description SBS1 in library LIB2.

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.
The Remove Community for SNMP (RMVCOMSNMP) command is used to remove a Simple Network Management Protocol (SNMP) community profile from the SNMP agent community list. The community profile consists of a community name, an object access specification, and a list of the SNMP managers that are part of the community. The community name combined with the ASCII community (ASCIIICOM) parameter defines a community.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM</td>
<td>Community name</td>
<td>Character value</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>ASCIIICOM</td>
<td>Translate community name</td>
<td>*YES, *NO</td>
<td>Optional</td>
</tr>
</tbody>
</table>

### Community name (COM)

Specifies the name of the SNMP community being removed. The community must already exist in the SNMP agent community list.

The possible values are:

`community-name`

Specify the name of the SNMP community being removed. The name may contain characters that cannot be displayed.

### Translate community name (ASCIIICOM)

Specifies whether the community name is translated to ASCII characters before it is compared with the community name specified in a request from an SNMP manager. This parameter is used in combination with the community name to determine the community to be removed.

The possible values are:

*YES

The community name is translated to ASCII characters before it is compared with a community name specified by an SNMP manager.

*NO

The community name is not translated to ASCII characters before it is compared with a community name specified by an SNMP manager.
Examples
RMVCOMSNMP COM(ROCHESTER)

This command removes community ROCHESTER from the SNMP agent community list.

Error messages

*ESCAPE Messages

TCP4001
Error occurred accessing SNMP configuration information.

TCP4009
Community does not exist.

TCP8050
*IOSYSCFG authority required to use &1.
Remove CRG Device Entry (RMVCRGDEVE)

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

The Remove Cluster Resource Group Device Entry (RMVCRGDEVE) command removes one or more configuration objects from a device cluster resource group. All configuration object entries can be removed but at least one configuration object entry must exist before the Start Cluster Resource Group (STRCRG) command can be called.

Ownership of the hardware associated with the configuration object being removed is not affected. The hardware is still owned by whatever node owned it before this command was called.

If an exit program is specified for the cluster resource group, the cluster resource group exit program is called with an action code of Remove Device Entry on all active nodes in the recovery domain. The cluster resource group status is set to Remove Device Entry Pending. If the exit program completes successfully, the cluster resource group status is reset to its value at the time the command was called. If the exit program fails and the cluster resource group cannot be restored to its original condition, the cluster resource group status is set to Indoubt.

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. Cluster Resource Services must be active on the node processing the request.
4. At least one node in the recovery domain must be active.
5. If the cluster resource group is Active, the last device entry cannot be removed. The cluster resource group must be ended first.

Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLUSTER</td>
<td>Cluster</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>CRG</td>
<td>Cluster resource group</td>
<td>Name</td>
<td>Required, Positional 2</td>
</tr>
<tr>
<td>CFGOBJ</td>
<td>Configuration object list</td>
<td>Values (up to 256 repetitions): Element list</td>
<td>Required, Positional 3</td>
</tr>
<tr>
<td></td>
<td>Element 1: Configuration object</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element 2: Configuration object type</td>
<td>*DEVD</td>
<td></td>
</tr>
</tbody>
</table>
**Cluster (CLUSTER)**
Specifies the cluster to which the cluster resource group belongs.

This is a required parameter.

*name* Specify the name of the cluster which contains the cluster resource group.

**Cluster resource group (CRG)**
Specifies the cluster resource group from which device list entries are being removed.

This is a required parameter.

*name* Specify the name of the cluster resource group which contains the device list entries.

**Configuration object list (CFGOBJ)**
Specifies the resilient devices that are to be removed from the cluster resource group.

You can specify up to 256 values for this parameter.

This is a required parameter.

**Element 1: Configuration object**

Specifies the auxiliary storage pool device description object which will be removed from the device list of the cluster resource group.

*name* Specify the name of the configuration object.

**Element 2: Configuration object type**

Specifies the object type of the configuration object specified for element 1.

*DEVD* Type of configuration object is a device description.

**Examples**

RMVCRGDEVE CLUSTER(MYCLUSTER) CRG(MYCRG)
CFGOBJ((IASP01 *DEVD))

This command removes configuration object IASP01 from an existing cluster resource group MYCRG in cluster MYCLUSTER.
Error messages

*ESCAPE Messages

CPF1999

Errors occurred on command.
Remove CRG Node Entry (RMVCRGNODE)

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

The Remove Cluster Resource Group Node Entry (RMVCRGNODE) command is used to remove a node from the recovery domain of a cluster resource group. The node being removed does not need to be active in the cluster to be removed from the recovery domain. When the node is removed from the recovery domain, the cluster resource group object is deleted from that system.

For primary-backup model, if the cluster resource group has no backup nodes in either the current recovery domain or the preferred recovery domain, the primary node cannot be removed.

This command results in the preferred recovery domain and the current recovery domain being updated.

This command will do the following for all cluster resource group types:
1. Set the cluster resource group status to Remove Node Pending.
2. Call the exit program on all active nodes in the recovery domain with an action code of Remove Node, if an exit program is specified for the cluster resource group.
3. Reset the cluster resource group status to the value at the time the command was called, if the exit program completes successfully on all nodes.
4. Set the cluster resource group status to Indoubt if the exit program fails on any node and the original state of the cluster resource group cannot be recovered.

This command will do the following for application cluster resource group:
1. If Cluster Resource Services configured the takeover IP address, it will be removed.

This command will do the following for device cluster resource groups:
1. If the node being removed is the current primary node, ownership of the devices specified in the cluster resource group are switched from the current primary to the new primary if none of the configuration objects on the current primary are varied on. If any configuration objects are varied on, an error message is returned. In addition, the new primary node must be active. Devices are not varied on after the ownership is switched. The node which is to become the new primary must be active in the cluster.

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. Cluster Resource Services must be started on the node processing the request.
4. At least one node in the recovery domain must be active.
5. For primary-backup model cluster resource groups:
   • The cluster resource group status must not be Active in order to remove the node that is currently the primary.
   • If the cluster resource group has no backup nodes, the primary node cannot be removed.
6. For peer model cluster resource groups, the last node designated as a peer node cannot be removed if the cluster resource group status is Active.
7. At least node must remain active in the recovery domain of the cluster resource group after the successful completion of the remove operation.

### Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Choices</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLUSTER</td>
<td>Cluster</td>
<td>Name</td>
<td>Required, Positional 1</td>
</tr>
<tr>
<td>CRG</td>
<td>Cluster resource group</td>
<td>Name</td>
<td>Required, Positional 2</td>
</tr>
<tr>
<td>NODE</td>
<td>Node identifier</td>
<td>Name</td>
<td>Required, Positional 3</td>
</tr>
</tbody>
</table>

### Cluster (CLUSTER)

Specifies the cluster to which the cluster resource group belongs.

This is a required parameter.

(name) Specify the name of the cluster which contains the cluster resource group.

### Cluster resource group (CRG)

Specifies the cluster resource group from which the node will be removed.

This is a required parameter.

(name) Specify the name of the cluster resource group.

### Node identifier (NODE)

Specifies the node that is to be removed from the recovery domain of the cluster resource group.

This is a required parameter.

(name) Specify the name of the node that is to be removed.

### Examples

RMVCRGNODE CLUSTER(MYCLUSTER) CRG(MYCRG) NODE(RCHCST03)

This command removes node RCHCST03 from the recovery domain of cluster resource group MYCRG in cluster MYCLUSTER.
Error messages

*ESCAPE Messages

CPF1999

Errors occurred on command.
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

© Copyright IBM Corp. 1998, 2006
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

Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.
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.
IBM Systems - iSeries: i5/OS Commands Starting with MRGTCPT (Merge TCP/IP Host Table)