

IBM

@server

iSeries

CL Commands Volume 18





@server

iSeries

CL Commands Volume 18

Contents

Command Descriptions	1
RST (Restore) Command Description	1
RSTAPARDTA (Restore APAR Data) Command Description	10
RSTAUT (Restore Authority) Command Description	11
RSTAUTBRM (Restore Authority Using BRM) Command Description	14
RSTCFG (Restore Configuration) Command Description	15
RSTDLOBRM (Restore DLO using BRM) Command Description	23
RSTDLO (Restore Document Library Object) Command Description	23
RSTLIB (Restore Library) Command Description	33
RSTLIBBRM (Restore Library Using BRM) Command Description	50
RSTLICPGM (Restore Licensed Program) Command Description	51
RSTOBJ (Restore Object) Command Description	57
RSTBRM (Restore Object Using BRM) Command Description	72
RSTOBJBRM (Restore Object Using BRM) Command Description	73
RSTS36F (Restore System/36 File) Command Description	74
RSTS36FLR (Restore System/36 Folder) Command Description	81
RSTS36LIBM (Restore System/36 Library Members) Command Description	86
RSTUSFCNR (Restore Ultimedia System Facilities Container) Command Description	93
RSTUSRPRF (Restore User Profiles) Command Description	95
RSMBKP (Resume Breakpoint) Command Description	104
RSMCTLRKY (Resume Controller Recovery) Command Description	104
RSMDEVRCY (Resume Device Recovery) Command Description	106
RSMHTTPCRL (Resume HTTP Crawl) Command Description	107
RSMLINRCY (Resume Line Recovery) Command Description	107
RSMNWIRCY (Resume Network Interface Recovery) Command Description	108
RSMRTVBRM (Resume Retrieve Using BRM) Command Description	109
RTVAUTLE (Retrieve Authorization List Entry) Command Description	109
RTVBCKUP (Retrieve Backup) Command Description	112
RTVBNDSRC (Retrieve Binder Source) Command Description	114
RTVCLSRC (Retrieve CL Source) Command Description	117
RTVCLNUP (Retrieve Cleanup) Command Description	119
RTVCFGSRC (Retrieve Configuration Source) Command Description	121
RTVCFGSTS (Retrieve Configuration Status) Command Description	124
RTVCURDIR (Retrieve Current Directory) Command Description	126
RTVDTAARA (Retrieve Data Area) Command Description	127
RTVDSKINF (Retrieve Disk Information) Command Description	131
RTVDOC (Retrieve Document) Command Description	132
RTVDLOAUT (Retrieve Document Library Object Authority) Command Description	137
RTVDLONAM (Retrieve Document Library Object Name) Command Description	141
RTVGRPA (Retrieve Group Attributes) Command Description	144
RTVJOBA (Retrieve Job Attributes) Command Description	148
Usage Notes	154
RTVJRNE (Retrieve Journal Entry) Command Description	156
RTVLIBD (Retrieve Library Description) Command Description	169
RTVMBRD (Retrieve Member Description) Command Description	171
RTVMSG (Retrieve Message) Command Description	176
RTVNETA (Retrieve Network Attributes) Command Description	181
RTVOBJD (Retrieve Object Description) Command Description	188
RTVPWRSCDE (Retrieve Power On/Off Schedule Entry) Command Description	198
RTVPDGPRF (Retrieve Print Descriptor Group Profile) Command Description	200
RTVPRD (Retrieve Product) Command Description	201
RTVPTF (Retrieve Program Temporary Fix) Command Description	203
RTVQMFORM (Retrieve Query Management Form) Command Description	205

RTVQMQRV (Retrieve Query Management Query) Command Description	207
RTVSFWPKGA (Retrieve Software Package Attributes) Command Description	209
RTVSWLSRC (Retrieve Stop Word List Source) Command Description	209
RTVSYINF (Retrieve System Information) Command Description	210
RTVSMGOBJ (Retrieve System Manager Object) Command Description.	211
RTVSYVAL (Retrieve System Value) Command Description	213
RTVS36A (Retrieve System/36 Attributes) Command Description	214
RTVTBLSRC (Retrieve Table Source) Command Description	216
RTVUSRPRF (Retrieve User Profile) Command Description	218
RTVWSCST (Retrieve Work Station Customizing Object Source) Command Description	225
RETURN (Return) Command Description	226
RVKACCAUT (Revoke Access Code Authority) Command Description	227
RVKOBJAUT (Revoke Object Authority) Command Description	228
RVKPUBAUT (Revoke Public Authority) Command Description	234
RVKUSRPMN (Revoke User Permission) Command Description	235
RVKWSOAUT (Revoke Workstation Object Authority) Command Description	237
ROLLBACK (Rollback) Command Description	240
RUNBCKUP (Run Backup) Command Description	241
Run Java (RUNJVA) Command Description	242
RUNLPDA (Run LPDA-2) Command Description	246
RUNQRY (Run Query) Command Description	252
AREXEC (Run Remote Command) Command Description	259
RUNRMTCMD (Run Remote Command) Command Description	259

Command Descriptions

RST (Restore) Command Description

RST Command syntax diagram

Purpose

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

For more information about integrated file system commands, see the Integrated file system topic in the File systems and management category of the Information Center.

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

For detailed restrictions on using this command to restore objects in libraries or to restore document library objects, see the Backup, Recovery, and Availability topic in the Information Center.

Required Parameter

DEV Specifies the path name of the device from which the objects are restored. For more information on specifying path names, refer to path names.

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

'diskette-device-path-name': Specify the path name of the diskette device used to restore the objects.

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

'tape-media-library-device-path-name': Specify the path name of the tape media library device used to restore the objects.

'tape-device-path-name': Specify the path name of the tape device used to restore the objects. A maximum of four tape devices can be specified.

Optional Parameters

OBJ Specifies the path name of the object to restore. You can specify a pattern for this path name. A maximum of 300 path names can be specified. For more information on specifying path names, refer to path names.

Additional information about name patterns is in the Integrated file system topic in the File systems and management category of the Information Center.

Element 1: Object Name

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

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

'object-path-name': Specify an object path name or a pattern that can match many names. If the default is specified in element 3, each component in the path name must exist with the exception of the last component. The object name in the last component is restored as new if it doesn't exist.

Element 2: Include or Omit

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 determines whether the subtrees are included or omitted.

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

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

Element 3: New Path Name

The third element specifies the new path name of the object.

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

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

SUBTREE

Specifies whether directory subtrees are included in the restore operation.

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

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

***NONE:** No subtrees are included in the restore operation. If a directory matches the object name pattern specified, the objects in the directory are included. If the directory has subdirectories, neither the subdirectories nor the objects in the subdirectories are included.

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

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

SYSTEM

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

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

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

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

SAVDATE

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

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

SAVTIME

Specifies the time the objects were saved. If this parameter is not specified, the version of the objects to be restored is the first version found on the volume.

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.

Notes:

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

OPTION

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

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

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

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

ALWOBJDIF

Specifies whether differences are allowed between the saved object and the restored object. The differences include:

- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.
- Primary Group: The primary group of an object on the system is different than the primary group of an object from the save operation.
- File level identifier: The file and/or member level identifier of a physical file on the save media is different than the file and/or member level identifier of the physical file on the system.
- Authorization list linking: The system on which an object with an authorization list is being restored is different from the system on which it was saved.

***NONE:** No differences are allowed between the saved object and the restored object. If the owner is different, the object is not restored. If the system is different for an object with an authorization list, the object is restored, but the object is not linked to its authorization list. If the primary group is different, the object is not restored. If the file and/or member level identifier of a physical file are different, the physical file data is not restored.

***ALL:** All differences are allowed between the saved object and the restored object. If the owner is different, the object is restored with the owner of the system on which it is restored. If the system is different for an object with an authorization list, the object is restored and linked to its authorization list. If the primary group is different, the object is restored with the primary group of the object on the system. If the file and/or member level identifiers are different, the file or member on the system will be renamed and the file or members is restored from the save media.

***OWNER:** The object owner can be different. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system. If owner differences are not allowed, the object is not restored.

***PGP:** The primary group can be different. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

***AUTL:** The system of an object with an authorization list can be different. The new object, which is being restored to a system that is different from which it was saved, is restored and linked to its authorization list. If the system of an object with an authorization list cannot be different, the object is restored but not linked to an authorization list.

FRCOBJCVN

Specifies whether to convert user objects to the format required for use in the current version of the operating system when the objects are restored.

Notes:

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

Element 1: Force Conversion

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

Note:

If this value is specified and the system value QFRCCVNRST has a value of "1," the restore operation proceeds as if FRCOBJCVN(*YES *RQD) is specified. If QFRCCVNRST has a value of "0", the restore operation proceeds as if FRCOBJCVN(*NO) is specified.

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

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

Note:

Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

Element 2: Objects to Convert

***RQD:** The objects are converted only if they require conversion to be used by the current operating system. If the objects are not observable, the objects are restored but are not converted.

***ALL:** All objects are converted regardless of their current format. Even the objects are in the current format, they are converted again. However, if the objects are not observable, the objects are not restored.

OBJID

Specifies whether the object ID of the restored object will be the object ID of the object from the

save media, the object ID of the object that exists on the system prior to the restore, or a new object ID generated by the system if the object does not exist on the system prior to the restore.

***SAVED:** The restored object will have the object ID it had when it was saved.

***SYS:** The restored object may have a new object ID generated by the system. If the object is being restored as a new object, a new object ID will be given to the restored object. If an object with the same name as the object from the media already exists on the system, the object ID of the restored object will be the same as the object ID of the object on the system before the restore.

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

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

***MOUNTED:** The objects are restored from the volumes placed in the device specified on the DEV parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

volume-identifier: Specify the identifiers of one or more volumes used to restore the objects.

LABEL

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

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

diskette-file-identifier: Specify the identifier (up to 17 characters) of the diskette file used for the restore operation.

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

SEQNBR

Specifies the tape file sequence number to be used.

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

sequence-number: Specify the sequence number of the file. Valid values range from 1 through 16777215.

ENDOPT

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

Note:

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

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

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

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

OPTFILE

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume. For more information on specifying path names, refer to path names.

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

*'optical-directory-path-name/**'*: The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

'optical-file-path-name': Specify the path name of the optical file.

OUTPUT

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

A stream file or user space is specified as a path name. For more information on specifying path names, refer to path names.

***NONE:** No output is created.

***PRINT:** Output information about the restore will be printed.

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

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

INFTYPE

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

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

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

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

Examples for RST**Example 1: Restoring All Data Not in Libraries or Document Library Objects**

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

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

Example 2: Restoring a Library

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

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

Example 3: Restoring All Files in MYLIB

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

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

Example 4: Restoring Everything Saved from the Home Directory

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

This command restores everything saved in the user's home directory from the tape device named TAP01.

Example 5: Restoring All Objects in the Current Directory

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

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

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

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

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

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

Example 6: Omitting Objects

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

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

Example 7: Renaming or Moving Objects

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

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

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

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

Example 8: Restoring From a Save File

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

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

Example 9: Using Symbolic Links

Assume the current directory contains the following symbolic links.

```
DevLink = /QSYS.LIB/TAP01.DEVD
```

```
DirLink = /SomeDirectory
```

```
FileLink = /SomeDirectory/SomeFile
```

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

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

To restore the objects in SomeDirectory from device TAP01:

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

Error messages for RST

*ESCAPE Messages

CPFA0DB

Object name not a QSYS object.

CPFA0DC

Object name not a QDLS object.

CPF370C

Not authorized to ALWOBJDIF parameter.

CPF3707

Save file &1 in &2 contains no data.

CPF3727

Duplicate device &1 specified on device name list.

CPF3738

Device &1 used for save or restore is damaged.

CPF3743

File cannot be restored, displayed, or listed.

CPF3768

Device &1 not valid for command.

CPF3782

File &1 in &2 not a save file.

CPF3794

Save or restore operation ended unsuccessfully.

CPF380D

Save or restore of entire system completed unsuccessfully.

CPF3805

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

CPF381E

Not authorized to ALWOBJDIF parameter.

CPF3812
Save file &1 in &2 in use.

CPF382A
Specified parameter not valid for QDLS file system.

CPF382B
Parameters not valid with multiple file systems.

CPF382C
OBJ parameter value not valid for QSYS file system.

CPF382D
Specified parameter not valid for QSYS file system.

CPF382F
OBJ parameter value not valid for QDLS file system.

CPF3823
No objects saved or restored.

CPF3826
*INCLUDE object required on OBJ parameter.

CPF3828
Error occurred while attempting to use &1.

CPF383A
Save or restore ended unsuccessfully.

CPF383B
End of file &1.

CPF383C
Storage limit exceeded for user profile &1.

CPF383D
Cannot use &1.

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

CPF3833
Specified value on DEV parameter not valid.

CPF3834
Too many values specified on the DEV parameter.

CPF3835
Tape devices do not support same densities.

CPF3839
&1 objects restored. &2 not restored.

CPF384A
Volume identifier &1 not valid.

CPF384B
Optical file specified not valid.

CPF384C
Error occurred during CCSID conversion.

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

CPF3840

Specified file for restore operation not found.

CPF5729

Not able to allocate object &1.

CPF9802

Not authorized to object &2 in &3.

CPF9825

Not authorized to device &1.

OPT1498

Volume name list exhausted on device &1.

OPT1502

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

OPT1605

Media or device error occurred.

RSTAPARDTA (Restore APAR Data) Command Description

Note: To use this command, you must have the 5722-SM1 (System Manager for iSeries) licensed program installed.

RSTAPARDTA Command syntax diagram

Purpose

The Restore APAR Data (RSTAPARDTA) command updates a problem log entry with the specified Authorized Problem Analysis Report (APAR) library name. This is the problem log entry saved with the APAR data on the service requester and is associated with the APAR library. If the problem log entry does not exist, one is created on the service provider. All spooled files saved using the Save APAR Data (SAVAPARDTA) command are respoiled on the service provider when RSTAPARDTA is run.

Parameter

FRMAPARLIB

Specifies the library from which to restore the APAR data. This library must contain the data from a specific APAR.

Example for RSTAPARDTA

```
RSTAPARDTA FRMAPARLIB(MYAPARLIB)
```

This command restores on the service provider the APAR data in library MYAPARLIB.

Error messages for RSTAPARDTA

*ESCAPE Messages

CPF2182

Not authorized to library &1

CPF39FA

Problem &1 &2 &3 not found

CPF39FB

Library &1 does not exist

CPF39FD

No APAR data restored from library &1

CPF39F2

Cannot allocate library &1

CPF39F5

Query of problem &1 failed

CPF39F9

Problem &1 &2 &3 in use

RSTAUT (Restore Authority) Command Description

RSTAUT Command syntax diagram

Purpose

The Restore Authority (RSTAUT) command restores the private authorities to user profiles. This command restores the same object use authority to specified objects in the user profile that each user profile had when all the profiles were saved by the Save System (SAVSYS) or Save Security Data (SAVSECDA) commands. It allows existing authorities, given after the save, to remain. Authority is not restored to the user profiles until the profiles are first restored to the system by the Restore User Profiles (RSTUSRPRF) command and all the objects (for which authority is being given) are restored to the same libraries where they are saved. The objects can be restored with the Restore Library (RSTLIB) or Restore Object (RSTOBJ), Restore Document Library Object (RSTDLO), Restore Configuration (RSTCFG), or Restore Objects in Directories (RST) commands.

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

1. Restore the operating system. This is an alternative method to load the program. This restores the QSYS library and ensures that the IBM-supplied user profiles are there.
2. Restore all the saved user profiles to the system (*ALL is the default for the USRPRF parameter) by using the RSTUSRPRF command.
3. Restore all the configuration and system resource management (SRM) objects to the system by using the RSTCFG command.
4. Restore all the user libraries (including the other IBM-supplied libraries) by using the RSTLIB command. Unless saved with the *IBM or *ALLUSR parameter, *NONSYS must be specified on the SAVLIB parameter to restore all user libraries in one operation.
5. Restore all document library objects to the system by using the RSTDLO command.
6. Restore all objects in directories using the RST command.
7. Restore the object authority to user profiles by using the RSTAUT command.

Note:

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

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

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

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

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. You must have *SAVSYS special authority to use this command.
3. Only one RSTAUT command can be run on a system at one time.

Optional Parameters

USRPRF

Specifies the names of one or more user profiles whose private authorities are being restored. The specified user profiles must first be restored using the RSTUSRPRF command.

***ALL:** Private authorities are being restored for all of the user profiles that are restored but do not have their private authorities restored. This includes user profiles that were restored using multiple RSTUSRPRF commands.

generic-user-profile-name:* Specify the generic name of the user profile. 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. For more information on the use of generic functions, refer to generic functions.

user-profile-name: Specify one or more names of specific user profiles. Both generic names and specific names can be specified in the same command.

» SAVASPDEV

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

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

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

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

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


auxiliary-storage-pool-device-name: Specify the name of the independent ASP device from which the private authorities were saved.

RSTASPDEV

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

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

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

auxiliary-storage-pool-device-name: Specify the name of the independent ASP device for which to restore the private authorities. 

Examples for RSTAUT

Example 1: Restore all authorities

```
RSTAUT
```

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

Example 2: Restore authorities for specific users

```
RSTUSRPRF USRPRF(USER1 USER2 USER3 USER4)
```

```
RSTLIB SAVLIB(USERLIB)
```

```
RSTAUT USRPRF(USER1 USER2 USER3)
```

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

Example 3: Restore authorities for specific libraries

```
RSTUSRPRF USRPRF(*ALL)
```

```
RSTLIB SAVLIB(USERLIBA)
```

```
RSTLIB SAVLIB(USERLIBB)
```

```
RSTLIB SAVLIB(USERLIBC)
```

```
RSTAUT USRPRF(*ALL)
```

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

Example 4: Restore authorities after multiple RSTUSRPRF commands

```
RSTUSRPRF USRPRF(USER1 USER2)
```

```
RSTLIB SAVLIB(USERLIBA)
```

```
RSTUSRPRF USRPRF(USER1 USER3)
```

```
RSTLIB SAVLIB(USERLIBB)
```

```
RSTAUT USRPRF(*ALL)
```

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

» **Example 5: Restore libraries and authorities to an independent ASP**

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

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

Error messages for RSTAUT

***ESCAPE Messages**

CPF2206

User needs authority to do requested function on object.

CPF3776

Not all user profiles had all authorities restored.

CPF3785

Not all subsystems ended.

CPF3855

RSTAUT not allowed at this time.

CPF386D

Prestart job failed.

» **CPF9814**

Device &1 not found. <<

» **CPF9833**

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

» **CPFB8ED**

Device description &1 not correct for operation. <<

RSTAUTBRM (Restore Authority Using BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.

RSTAUTBRM syntax diagram

Purpose

» The Restore Authority using BRM (RSTAUTBRM) command re-establishes the object owner, primary group and authorization list names for objects on basic user auxiliary storage pools (2-32) if this information was saved. «

» This information is saved unless you have explicitly omitted it from the save by adding a *USRASPAUT entry to either the *SAVSYS or *SECDTA entry in the backup policy omit list. «

The user profiles must be restored before using the RSTAUTBRM command.

The libraries and objects on the saved user auxiliary storage pools must be restored before using the RSTAUTBRM command.

Note:

» The RSTAUTBRM command does not replace the OS/400 RSTAUT command. The RSTAUTBRM command is used in addition to the RSTAUT command when restoring the authority information of saved basic user auxiliary storage pool objects during a system recovery. «

Example for RSTAUTBRM

» **Example 1: Restoring saved authority information for all ASPs** «

```
RSTAUTBRM  
or  
RSTAUTBRM USRASP(*ALLUSR)
```

» In this example, owner profile, primary group, and authorization list names for objects for all basic user auxiliary storage pools is restored. «

» **Example 2: Restoring saved authority information for ASPs 2 and 3** «

```
RSTAUTBRM USRASP(2 3)
```

» In this example, owner profile, primary group, and authorization list names for objects contained in basic user auxiliary storage pools 2 and 3 are restored. «

Error messages for RSTAUTBRM

No error messages.

RSTCFG (Restore Configuration) Command Description

RSTCFG Command syntax diagram

Purpose

The Restore Configuration (RSTCFG) command restores to the system a configuration object that was saved by the Save System (SAVSYS) or Save Configuration (SAVCFG) command. The types of objects that can be restored by this command are listed in the OBJTYPE parameter.

The user profile of the system default owner (QDFTOWN) becomes the default owner of any objects being restored on the system when the profile of the owner is not known to the system.

If an object already exists, in the library to which the object is restored, the public and private authorities of the existing object are kept. If the object does not exist in the library, all public authorities are restored, but any private authorities must be granted again.

Restrictions:

1. To use this command, the user must have *SAVSYS authority, or object existence authority for (or be the owner of) each object specified if the object already exists on the system.
2. This command is shipped with public *EXCLUDE authority.
3. The device configuration object must be varied off when it is being restored. To vary off a device configuration object, use the Vary Configuration (VRYCFG) command.
4. With the exception of overrides for the output listing file, this command ignores all file overrides currently in effect for the job.
5. System resource management (SRM) objects are not restored if the RSTCFG command is run using media that was created prior to V2R2M0.
6. If the RSTCFG command and the SAVSYS or SAVCFG commands are not run on the same system, the configuration objects may not match the physical hardware on the target system.
7. If you restore system resource management objects on a system other than the one on which the SAVSYS or SAVCFG command was saved, the system then treats the target system hardware as new and creates all new resource names, making the existing configuration descriptions useless. If this occurs, you need to restore the correct system resource management database from the most current SAVSYS or SAVCFG command for that command. If neither of these is available, you must change existing configuration descriptions to reflect the new resource names.

Required Parameters

OBJ Specifies the names of one or more configuration objects being restored.

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

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

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

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

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

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

***SAVF:** The save file specified on the SAVF parameter is used for the restore operation.

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

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

tape-device-name: Specify the names of one or more tape devices used for the restore operation. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order in which they are used.

The device name must already be known on the system by a device description. If using more than one tape drive (up to four), specify the names of the drives in the order they are used. If the

objects are on more than one tape volume, the user may want to use more than one tape drive, so that one tape volume can be rewound and unloaded while another tape drive processes the next volume.

Optional Parameters

OBJTYPE

Specifies the types of OS/400 system objects that are restored.

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

object-type: Specify the value for each of the types of objects that are being restored. The following types can be specified:

- *CFGL** Configuration list
- *CNL** Connection list
- *COSD** Class-of-service description
- *CTLD** Controller description
- *DEVD** Device description
- *LIND** Line description
- *MODD** Mode description
- *NTBD** NetBIOS description
- *NWID** Network interface description
- *NWSD** Network server description

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

Note: *SRMSPC are also saved, but not restored as an object type. Specify *SRM on the OBJ parameter to restore SRM data.

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

Note: The version of the objects that is restored is the first version found in the specified location.

***MOUNTED:** The objects are restored from the volumes placed in the device specified on the DEV parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are placed in the device and used to restore the objects.

SEQNBR

Specifies which sequence number is used for the restore operation.

***SEARCH:** The volume placed in a device is searched for a data file containing the saved device configuration objects. When a match is found, the configuration objects are restored. If the last operation on the device specifies ENDOPT(*LEAVE), the tape remains positioned at the location where the last operation ended. The file search starts with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last operation, or if the tape was manually rewound after an ENDOPT(*LEAVE) operation, the search starts with the first data file on the volume.

file-sequence-number: Specify the sequence number of the file. Valid values range from 1 through 16777215.

ENDOPT

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

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

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

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

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

SAVF Specifies the qualified name of the save file used to restore the save data.

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

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

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

library-name: Specify the name of the library to be searched.

save-file-name: Specify the name of the save file.

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

Attention:

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

***ALL:** All SRM information is restored.

***NONE:** No SRM information is restored.

***HDW:** All hardware information is restored.

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

ALWOBJDIF

Specifies whether certain differences encountered during a restore operation are allowed. There are three differences that apply to the RSTCFG command:

- The owner of the object on the system is different than the owner of the object from the save operation.
- The object is secured by an authorization list and is being restored to a system other than the one on which it was saved.
- The primary group of the object on the system is different than the primary group of the object from the save operation.

Note:

The user of this parameter must have *ALLOBJ authority.

***NONE:** The differences described above are not allowed for the restore operation. For an ownership difference, the object is not restored. For an authorization list difference, the object is restored, but the object is not linked to the authorization list, and public authority is set to *EXCLUDE. For a primary group difference, the object is not restored.

***ALL:** The differences described above are allowed for the restore operation. The object is restored.

Notes:

1. If the owners of the object do not match, the object is restored, but it retains the ownership and authorities it had before the restore operation.
2. The informational message causes a diagnostic message to be sent indicating that security or integrity changes occurred during the restore operation. This results in an escape message for the restore operation.
3. If the user is restoring objects to a system other than the one on which they were saved and the objects are secured by an authorization list, specifying *ALL automatically relinks the objects to the authorization list. If the authorization list does not exist on the new system, a message that includes the name of the missing list is issued.
4. If the primary groups of the object do not match, the object is restored, but it retains the primary group it had before the restore operation.
5. The informational message causes a diagnostic message to be sent indicating that security or integrity changes occurred during the restore operation. This results in an escape message for the restore operation.

OUTPUT

Specifies whether a listing that shows information about the status of the object is created and directed to an output file. The listing shows the restore information and shows all objects, restored,

not restored, and excluded. Information about each object's security is listed for the restored objects. More information on this parameter is in commonly used parameters.

***NONE:** No output is created.

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

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

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

OUTFILE

Specifies the qualified name of the database file to which the information about the object is directed when *OUTFILE is specified on the OUTPUT parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QASRRSTO in QSYS with the format name QSRRST as a model.

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

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

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

library-name: Specify the name of the library to be searched.

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

OUTMBR

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

Element 1: Member to Receive Output

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

member-name: Specify the name of the file member that receives the output. If OUTMBR(*member-name*) is specified and the member does not exist, the system creates it. If the member exists, the user can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Operation to Perform on Member

***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.

OPTFILE

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume. For more information on specifying path names, refer to path names.

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

*'optical-directory-path-name/***: The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

Examples for RSTCFG

Example 1: Restoring All Objects

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

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

Example 2: Restoring a Device Description

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

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

Example 3: Restoring a Network Server Description

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

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

Error messages for RSTCFG

*ESCAPE Messages

CPF26C1

Network server &1 not varied off.

CPF370C

Not authorized to ALWOBJDIF parameter.

CPF3707

Save file &1 in &2 contains no data.

CPF3709

Tape devices do not support same densities.

CPF3727

Duplicate device &1 specified on device name list.

CPF3728

Device &1 specified with other devices.

CPF3733

&2 &1 in &3 previously damaged.

CPF3738

Device &1 used for save or restore is damaged.

CPF3743

File cannot be restored, displayed, or listed.

CPF3748

Object information for library &1 damaged.

CPF376B

File &1 not found.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF3769
File found on media not save/restore file.

CPF377B
No objects restored. &3 objects excluded.

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

CPF3780
Specified file for library &1 not found.

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

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

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

CPF3793
Machine storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3796
Storage limit exceeded for user profile &4.

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

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

CPF3812
Save file &1 in &2 in use.

CPF3883
S/38 device configuration objects not restored.

CPF5729
Not able to allocate object &1.

CPF9809
Library &1 cannot be accessed.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

RSTDLOBRM (Restore DLO using BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.

RSTDLOBRM Command syntax diagram

Purpose

The Restore DLO using BRM (RSTDLOBRM) command restores documents, folder and distribution objects (mail) that were saved to tape or to save files. The media policy in force when the media information was saved determines whether this information resides on tape or is stored in save files.

Note: Using the RSTDLOBRM command allows you to restore selected copies of document, folder and distribution objects to the system based on the number of copies that are in the BRMS media content information. This is accomplished by specifying a copy number for the SAVLVL prompt.

Restrictions:

1. To use this command, you must have *SAVSYS or *ALLOBJ special authority or be enrolled in the system distribution directory.
2. You cannot run another RSTDLOBRM, RSTDLO, SAVDLOBRM, SAVDLO or Reclaim Document Library Object (RCLDLO) command specifying DLO(*ALL) while this command is running.

Notes:

1. Do not precede an entry with an asterisk unless that entry is a “special value” that is shown (on the display itself or in the help information) with an asterisk.
2. You can restore data from a TSM server device by using this command. You can only specify one TSM device or *MEDCLS, which must select a TSM device. The TSM device selected can either be *APPC, which supports the SNA network protocol, or *NET, which supports the TCP/IP protocol.

Note: This command should not be used by control group *EXIT item processing as results will be unpredictable.

Example for RSTDLOBRM

Example 1: Restoring All DLO in a Subfolder

```
RSTDLOBRM DLO(*ALL) SAVFLR('ACCTMST/ACCT96')  
COPY(*CURRENT) DEV(TAP09)
```

This command allows you to select and restore folders and subfolders that were saved using the SAVDLOBRM command and whose contents information still exist in the media contents information. In the example the restore operation is for all DLO with a folder called ACCTMST and a subfolder called ACCT96. The most recent copy is restored using device TAP09.

Error messages for RSTDLOBRM

None

RSTDLO (Restore Document Library Object) Command Description

RSTDLO Command syntax diagram

Purpose

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

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

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

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

Note:

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

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

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

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

If a document is restored that had a mail log entry when it was saved, the mail log entry is restored if the distribution tracking object exists on the system. If the distribution tracking object does not exist on the system, a message is sent saying that the document was restored without a mail log entry.

If this command ends abnormally, objects are left on the system in an unknown state and cannot be found in a library. This can happen if a power failure occurs when this command is run. The Reclaim Storage (RCLSTG) command can be used to clean up the auxiliary storage and delete most of those objects from the system; however, unknown mail objects are not cleaned up with the RCLSTG command.

When a set of documents and folders are restored, all documents and folders in the set must exist in the same diskette, tape, optical volume, or save file.

If a document exists in more than one tape or diskette file, the user can control which document is restored by specifying the media file using the sequence number (tape) or label (tape or diskette) parameter. If more than one version of the document exists, the SAVDATE and SAVTIME parameters can also be used to select the correct document.

When text search services are on the system and the user restores a document library object, the text search index for the object is restored.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. To use this command, you must have *SAVSYS or *ALLOBJ special authority or be enrolled in the system distribution directory.
3. This command cannot be run when RCLDLO DLO(*ALL) is running because RCLDLO requires exclusive use of internal objects.
4. When saving or restoring to an existing database file using the OUTFILE parameter, the user must have execute authority to the output file library.

Required Parameters

DLO Specifies the document library objects to be restored.

***ALL:** All documents, folders, and distribution objects (mail) that are saved on the media and meet the values specified on the SAVFLR parameter are restored.

***MAIL:** All distribution objects and documents that were referred to by a mail log are restored.

***SYSOBJNAM:** The documents with the system object names specified in the SYSOBJNAM parameter are restored.

document-name: Specify the user-assigned names of the documents to be restored. Up to 12 characters can be specified for the name; up to 300 document names can be specified. All documents named must be in the folder specified on the SAVFLR parameter.

DEV Specifies the name of the device to be used to restore the document library objects. The device name must be known on the system by a device description.

***SAVF:** The document library objects are restored from a save file.

diskette-device-name: Specify the name of the diskette device to be used to restore the document library objects.

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

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

tape-device-name: Specify the names of one or more tape devices used for the restore operation. If more than one tape device is used, specify the names of the devices in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Optional Parameters

SAVFLR

Specifies the name of the folder on the media from which the documents and folders are restored.

***ANY:** All document library objects that meet the values specified on the DLO parameter are restored, regardless of the folders (if any) from which they were saved. This value is valid only if *ALL, *MAIL, or *SYSOBJNAM is specified on the DLO parameter.

***NONE:** The documents to be restored were saved as documents without folders.

Note: *NONE is valid only when DLO(*ALL) is specified.

saved-folder-name: Specify the name of the saved folder from which documents or folders are to be restored. Up to 63 characters can be specified for the folder name. When DLO(*ALL) is specified, up to 300 folder names can be specified. The name of a saved folder must be specified when DLO(*document-name*) is specified.

RENAME

Specifies the new user-assigned name for the restored document.

***SAME**: The documents being restored have the same user-assigned names that they have on the media.

new-document-name: Specify the new document names that the documents have after they are restored. Up to 300 user-assigned names can be specified for documents being restored.

RSTFLR

Specifies the name of the folder in which the folders and documents to be restored will be placed. The name can be any fully-qualified folder name up to 63 characters. The folder named is not required to exist if DLO(*ALL) is specified and the media contains the folder specified on the SAVFLR parameter.

***SAME**: The documents to be restored are placed into the same folder from which they were saved.

restore-folder-name: Specify the name of the folder in which the restored documents and folders are to be placed.

SYSOBJNAM

Specifies the system object name of the documents to be restored. This parameter is valid only when DLO(*SYSOBJNAM) is specified.

***NONE**: A system object name is not specified.

system-object-name: Specify the 10-character system object names of the documents to be restored. A maximum of 300 names can be specified.

VOL

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

***MOUNTED**: The objects are restored from the volumes placed in the device specified on the DEV parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

Note:

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

volume-identifier: Specify the identifiers of one or more volumes in the order they are placed in the device and used to restore the objects.

SEQNBR

Specifies the tape sequence numbers used for the restore operation. This parameter is valid for tape only.

***SEARCH**: The tape is searched for the first data file with an identifier matching the LABEL parameter value and with contents of a minimum of one of the specified document library objects. If the last operation on the device specified ENDOPT(*LEAVE) (that is, the tape is positioned at the location at which the last operation ended), the file search begins with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not specified on the last operation (or if the tape has been rewound since an ENDOPT(*LEAVE) operation), the search begins with the first data file on the volume.

Note:

If a folder is contained in more than one file or on more than one tape, a message is sent indicating that the folder may not have been fully restored. To restore the rest of the folder, run the RSTDLO command again specifying the next sequence number.

Element 1: Beginning Sequence Number

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

Element 2: Ending Sequence Number

***ONLY:** The ending sequence number is the same as the beginning sequence number.

ending-sequence-number: Specify the sequence number of the last file used for the restore operation. Valid values range from 1 through 16777215.

ENDOPT

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

Note:

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

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

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

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

LABEL

Specifies the file label that is used to find the file that was written to the media during the save operation.

***GEN:** The system generates the default name of the file label for which to search.

data-file-identifier: Specify the media data file identifier, which includes up to 17 alphanumeric characters, of the file that contains the documents and folders to be restored.

SAVF Specifies the qualified name of the save file that contains the saved document library objects to be restored.

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

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

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

library-name: Specify the name of the library to be searched.

save-file-name: Specify the name of the save file from which to restore DLOs.

NEWOBJ

Specifies whether a new library-assigned name is created for the folder or document being restored.

***SAME**: The library-assigned name is restored to the name used when the document library objects were saved.

***NEW**: A new library-assigned name is created for each document or folder being restored. If ***NEW** is specified on this parameter, then **ALWOBJDIF(*ALL)** cannot be specified.

SAVDATE

Specifies the date on which the document library objects were saved. If more than one version of the document library objects exist on the media, use this parameter to identify which version of the document library objects to restore. The date must be expressed in the job date format; if separators are used, the value must be enclosed in apostrophes. If the **SAVDATE** parameter is not specified, the version of the documents and folders to be restored will be the first version found on the volume or the version found with the specified file label.

SAVTIME

Specifies the time when the document library objects were saved. If more than one version of the document library objects exist on the media with the same value for the date saved, use this parameter to identify which version of the document library objects to restore. Express the time as a 6-digit value, in the format hours, minutes, seconds (hhmmss). If separators are used, the value must be enclosed in apostrophes ('hh:mm:ss'). If a volume identifier is specified, but **SAVTIME** is not specified, the version of the document library objects to be restored will be the first version found on the volume or the version found with the specified file label.

This parameter is valid only if **SAVDATE** is also specified.

ALWOBJDIF

Specifies whether the following differences encountered during the **RSTDLO** operation are allowed:

- The owner of the object on the system is different than the owner of the object from the save.
- The primary group of the object on the system is different than the primary group of the object from the save.
- The system object name on the system does not match the system object name on the tape or diskette being restored.
- The object is secured by an authorization list and is being restored to a system other than the one on which it was saved.

The **ALWOBJDIF** parameter can be used to allow an object to be restored whose owner or object name on the system is different than on the media used for the restore operation. By specifying the ***ALL** special value, an object with a different name is restored to the name on the media, while an object with a different owner keeps the owner name from the system instead of the media.

Note:

In order to use this parameter, ***ALLOBJ** authority is required.

***NONE**: The differences described above are not allowed on the restore operation. For authorization list cases, the object is restored, but the object is not linked to the authorization list, and public authority is set to ***EXCLUDE**. For other cases, a diagnostic message is sent for the object, and the object is not restored.

***ALL:** All the differences described above are allowed for the restore operation. The object is restored.

Notes:

- If the owners of the object do not match, the object is restored, but it keeps the ownership and authorities of the object on the system before the restore operation.
- If the primary groups of the object do not match, the object is restored, but it keeps the primary group of the object on the system before the restore operation.
- The informational message flags a diagnostic message to be sent indicating that security or integrity changes occurred during the restore operation. This results in an escape for the restore function.
- If *ALL is specified on this parameter, then NEWOBJ(*NEW) cannot be specified.
- If the user is restoring objects to a system different from the one on which they were saved and the objects are secured by an authorization list, specifying *ALL automatically links the objects to the authorization list. If the authorization list does not exist on the new system, a message that includes the name of the missing list is issued.

SAVASP

Specifies the number of the auxiliary storage pool (ASP) on media from which saved documents and folders are to be restored.

***ANY:** The documents and folders saved in any ASP are restored.

auxiliary-storage-pool-number: Specify the number of the ASP from which documents and folders are restored.

RSTASP

Specifies the number of the auxiliary storage pool (ASP) on media in which restored documents and folders are to be placed.

***SAVASP:** The documents and folders are placed in the same ASP from which they were saved.

auxiliary-storage-pool-number: Specify the number of the ASP in which restored documents and folders are placed.

OUTPUT

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

***NONE:** No output is created.

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

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

Note:

You must specify the database file name on the OUTFILE parameter when OUTPUT(*OUTFILE) is specified.

OUTFILE

Specifies the qualified name of the database file to which the information about the object is directed when *OUTFILE is specified on the OUTPUT parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QAOJRSTO in QSYS with the format name QOJRST as a model.

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

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

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

library-name: Specify the name of the library to be searched.

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

OUTMBR

Specifies the name of the database file member to which the output is directed. If a member already exists, the system uses the second element of this parameter to determine whether the member is cleared before the new records are added. If the member does not exist and a member name is not specified, the system creates a member with the name of the output file specified on the OUTFILE parameter. If an output file member name is specified, but the member does not exist, the system creates it.

Element 1: Member to Receive Output

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

member-name: Specify the name of the file member that receives the output. If OUTMBR(*member-name*) is specified and the member does not exist, the system creates it. If the member exists, the user can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Operation to Perform on Member

***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.

OPTFILE

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume. For more information on specifying path names, refer to path names.

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

'optical-directory-path-name/':* The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

'optical-file-path-name': Specify the path name of the optical file.

Examples for RSTDLO

Example 1: Restoring Documents with System Object Names

```
RSTDLO DLO(*SYSOBJNAM) DEV(TAP01)
      SYSOBJNAM(HZ83B55219)
```

This command restores the document named HZ83B55219 from the tape unit TAP01.

Example 2: Restoring Documents from a Save Folder

```
RSTDLO DLO(A) DEV(diskette-device-name) SAVFLR(X)
```

This command restores the document named A from folder X.

Example 3: Restoring All Documents

```
RSTDLO DLO(*ALL) DEV(TAP01)
```

This command restores all documents and folders that are on the first tape file on tape unit TAP01.

Example 4: Restoring a Folder Saved from the System ASP to a User ASP

```
RSTDLO DLO(*ALL) FLR(Y) SAVASP(1) RSTASP(2)
```

This command restores folder Y, which was saved from ASP 1, to user ASP 2. Folder Y must be deleted from ASP 1 before it can be restored to ASP 2.

Example 5: Creating New Library-Assigned Name

```
RSTDLO DLO(*SYSOBJNAM) DEV(TAP01)  
      SYSOBJNAM(HZ83B55219) NEWOBJ (*NEW)
```

This command restores document HZ83B55219 from tape unit TAP01 and gives it a new library-assigned name and a new system object name.

Example 6: Renaming Documents

```
RSTDLO DLO(A B) DEV(TAP01) SAVFLR(C) RENAME(Y Z)  
      RSTFLR(X)
```

This command restores documents A and B from within folder C. Document A is renamed to Y and document B is renamed to Z. The command then puts them in folder X.

Example 7: Specifying Sequence Numbers

```
RSTDLO DLO(*ALL) DEV(tape-device-name) SAVFLR(A)  
      SEQNBR(1 3) LABEL(*GEN)
```

This command restores all of folder A from tape files with the sequence numbers 1, 2, and 3, and the label QDOC or QDOCxxxx.

Example 8: Specifying Allowed Differences

```
RSTDLO DLO(A) DEV(TAP01) SAVFLR(X)  
      ALWOBJDIF(*ALL)
```

This command restores document A from folder X. If document A in folder X exists on the system and the owner of the document on the system does not match the owner of the document being restored, the document is restored and the owner of the document on the system remains unchanged.

Example 9: Reporting Information about Objects Restored and Not Restored

```
RSTDLO DLO(*ALL) DEV(TAP01) OUTPUT(*OUTFILE)  
      OUTFILE(INFO92) OUTMBR(FOURQT *ADD)
```

This command restores all documents and folders from the tape device TAP01. A list reporting information about objects restored and objects not restored is directed to the output file INFO92. The output is received in the member FOURQT as an addition to existing information in the member.

Error messages for RSTDLO

*ESCAPE Messages

CPF370C

Not authorized to ALWOBJDIF parameter.

CPF3718

Restore command not valid for file &1.

CPF3728
Device &1 specified with other devices.

CPF3767
Device &1 not found.

CPF3780
Specified file for library &1 not found.

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

CPF381B
No DLOs restored to ASP &1.

CPF3812
Save file &1 in &2 in use.

CPF384D
Save or restore operation not allowed on ASP &1.

CPF8AB5
ASP &5 is not configured.

CPF8A47
Internal system objects in use.

CPF90AF
RSTFLR value not allowed.

CPF90A4
RENAME value not allowed.

CPF90B4
&1 folders restored to system, &2 not restored.

CPF90CD
Not authorized to restore distributions.

CPF90CF
Search index data base is damaged.

CPF90E0
Not enough authority for ALWOBJDIF(*ALL).

CPF90E7
Document library objects not restored.

CPF9003
&1 document library objects restored. &10 not restored.

CPF905C
Error occurred trying to find a translation table.

CPF9050
Ending sequence number not valid.

CPF9069
User not permitted to restore into folder &1.

CPF908A
Requester &1 not enrolled.

CPF909B
&1 document library objects restored. &10 not restored.

CPF9412
List of folder names not allowed with DLO parameter.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9831
Cannot assign device &1.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

CPF9850
Override of printer file &1 not allowed.

CPF9851
Overflow value for file &1 in &2 too small.

CPF9860
Error occurred during output file processing.

CPF9899
Error occurred during processing of command.

OPT1498
Volume name list exhausted on device &1.

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

OPT1605
Media or device error occurred.

RSTLIB (Restore Library) Command Description

RSTLIB Command syntax diagram

Purpose

The Restore Library (RSTLIB) command restores to the system one library or a group of libraries that was saved by the Save Library (SAVLIB) command. The RSTLIB command restores the whole library, which includes the library description, object descriptions, and contents of the objects in the library.

For job queues, message queues, output queues, data queues, and logical files, only the object descriptions are restored, because only the definitions are saved. Also, logical file access paths may be restored if they were saved. More information on restoring access paths is in the Database Programming topic in the Information Center.

This command can be used to restore libraries whose objects had their storage freed by the corresponding SAVLIB command of the restore operation, or libraries deleted by the Delete Library (DLTLIB) command. If the data portions of the objects in the saved libraries were not freed, each library is copied into the same area of storage that it previously occupied. If the storage was freed, the system finds the needed storage to store the library contents (the object description and data portion of every file, module, program, service program, Structured Query Language (SQL) package, and journal receiver in the library). If the library does not exist on the system because it has been deleted or is being restored on a different system, the system must find the storage for everything that is in the library, including the library description.

When the owner profile does not exist on the system, the user profile of the system default owner (QDFTOWN) becomes the default owner of any object being restored in the system.

If an object already exists in the library in which it is being restored, the public and private authorities of the existing object are retained. If the object does not exist in the library, all public authorities are restored, but private authorities must be granted again. For an existing output queue object that is actively spooling during the restore operation, or a data queue that already exists in the library, the object is not restored, and a diagnostic message is sent.

If an object is being restored over an existing object on the system, the object auditing value of the existing object is kept. If the object is being restored as new to the system, the object auditing value is restored from the media. Additionally, if the object is a library, the default auditing value for each object created in the library is restored if the library is being restored as new; otherwise, the default auditing value is restored from the media.

Notes:

1. To restore a save file to a library where it does not already exist, a user must have authority to the Create Save File (CRTSAVF) command.
2. The RSTLIB command ignores all file overrides currently in effect for the job, except the overrides for the restore output file.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The user of this command must have *SAVSYS authority specified in the user profile by the SPCAUT parameter, or must have all of the following:
 - a. Read and add authority for, or be the owner of, each library specified.
 - b. Object existence authority for, or be the owner of, each object in the library if the object already exists in the library on the system. Object existence and use authority are required for message queue objects. If the object does not exist, the user must have add authority for the user profiles that own each object being restored. If the user does not have the correct authority for all the libraries and objects specified, only those for which the authority has been given are restored.
 - c. If VOL(*SAVVOL) is specified, use authority to the saved-from library.
 - d. Use authority for the save file is required when restoring libraries from a save file. Use authority for the device description and the device file are required when restoring libraries from a tape, a diskette, or an optical device.
3. When using a media definition, the user must have *USE authority to the media definition and *EXECUTE authority to the media definition library.

4. The current versions of programs on the system should not be run while the library that contains those programs is being restored. If any program is running while it is being restored, the program will not be restored.
5. When saving or restoring to an existing database file using the OUTFILE parameter, the user must have execute authority to the output file library.

Required Parameters

SAVLIB

Specifies the name of the library, or the *NONSYS, *ALLUSR, or *IBM group of libraries, to be restored on the system. You cannot specify *NONSYS, *ALLUSR, or *IBM if RSTLIB(*SAVLIB) is not specified.

Note:

For values *NONSYS and *IBM, all other operations on the system must be ended before this option is specified. This requires ending all subsystems through the End Subsystem (ENDSBS(*ALL)) command or End System (ENDSYS) command.

***NONSYS:** Libraries saved by the Save Library (SAVLIB) command with LIB(*NONSYS) specified are restored.

You can do a RSTLIB SAVLIB(*IBM) and a RSTLIB SAVLIB(*ALLUSR) from a SAVLIB LIB(*NONSYS).

» ***ALLUSR:** 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	» QSYS2xxxx«	QUSROND
QGPL	QS36F	QUSRPOSGS
QGPL38	QUSER38	QUSRPOSSA
QMPGDATA	QUSRADSM	QUSRPYMSVR
QMQMDATA	QUSRBRM	QUSRDRARS
QMQMPROC	QUSRDIRCL	QUSRSYS
QPFRDATA	QUSRDIRDB	QUSRVI
QRCL	QUSRIJS	QUSRVxRxMx
» QRCLxxxx«	QUSRINFSKR	
» QSYS2«	QUSRNOTES	

Notes:

1. » 'xxxxx' is the number of a primary auxiliary storage pool.«
2. A different library name, of the form QUSRVxRxMx, can be created by the user for each release that IBM supports. VxRxMx is the version, release, and modification level of the library.

***IBM:** Restores all system (IBM) libraries except for the following: >

QDOC	QRCLxxxxx	QSYS2	QUSRINFSKR
QDOCxxxx	QRCYxxxxx	QSYS2xxxxx	QUSRNOTES
QDSNX	QRECOVERY	QS36F	QUSROND
QGGL	QRPLOBJ	QTEMP	QUSRPOSGS
QGGL38	QRPLxxxxx	QUSER38	QUSRPOSSA
QMPGDATA	QSPL	QUSRADSM	QUSRPYMSVR
QMOMDATA	QSPLxxxx	QUSRBRM	QUSRDRARS
QMOMPROC	QSRV	QUSRDIRCL	QUSRSYS
QPFRDATA	QSYS	QUSRDIRDB	QUSRVI
QRCL	QSYSxxxxx	QUSRIJS	QUSRVxRxMx



Note:

A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

The following libraries with names that do not begin with the letter Q are also restored:

#CGULIB	#DSULIB	#SEULIB
#COBLIB	#RPLIB	
#DFULIB	#SDALIB	

library-name: Specify the name of the library to restore. The library name must be the same name that was used when the library was saved.

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

***SAVF:** The restore operation is done by using the save file specified on the SAVF parameter.

***MEDDFN:** The restore operation is done using the devices and media identified in the media definition specified by the Media Definition (MEDDFN) parameter.

diskette-device-name: Specify the name of the diskette device that is used for the restore operation.

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

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

tape-device-name: Specify the names of one or more tape devices used for the restore operation. If more than one tape device is specified, they are used serially in the order in which they are specified. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Optional Parameters

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

***MOUNTED:** The library is restored from the volumes that are currently placed in the device specified on the DEV parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command. The first version of the saved library that is found on the media is restored, unless a specific version is identified by the SAVDATE parameter and SAVTIME parameter, or for tape, the SEQNBR parameter.

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

***SAVVOL:** The system, by using the save or restore history information, determines which volumes contain the most recently saved version of the library. If the device type specified in the DEV parameter does not match the device type of the most recently saved version of the library, an error message is sent to the user, and the function is ended. If *SAVVOL is specified, the parameters SAVDATE and SAVTIME cannot be specified. If *SAVVOL is specified, *SEARCH must be specified on the SEQNBR parameter.

Note: If this is a restore from an optical device, only the first 6 characters of the volume name are available. If the volume name of the optical media exceeds 6 characters, the volume may not be found. You should specify the complete volume or file name on the command instead of using VOL(*SAVVOL) when the name of the optical media exceeds 6 characters.

volume-identifier: Specify the identifiers of one or more volumes in the order they are placed in the device and used to restore the libraries.

SEQNBR

Specifies, only when tape is used, which sequence number is used for the restore operation.

***SEARCH:** The tape is searched for a data file with an identifier that matches the LABEL parameter value; when a match is found, the data file is restored. If the last operation on the device specified ENDOPT(*LEAVE), the tape is positioned at the location where the last operation ended, and the file search starts with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last operation, or if the tape was manually rewound after an ENDOPT(*LEAVE) operation, the search starts with the first data file on the volume.

file-sequence-number: Specify the sequence number of the file. Valid values range from 1 through 16777215.

If *NONSYS, *ALLUSR, or *IBM is specified on the SAVLIB parameter, the sequence number specifies the location of the file QFILE. The QFILE file is at the beginning of the *NONSYS, *ALLUSR, or *IBM save operation. The QFILE file contains the list of libraries saved.

LABEL

Specifies the name that identifies the data file on the tape or diskette used for the restore operation. This must be the label specified on the save command.

***SAVLIB:** The file label is the name of the library specified on the SAVLIB parameter.

data-file-identifier: Specify the data file identifier (up to 17 characters) of the data file used for the restore operation. This option is valid only for single library restore operations.

ENDOPT

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

Note:

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

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

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

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

STRLIB

Specifies the name of the starting library for a *NONSYS, *IBM, or *ALLUSR restore operation.

If an irrecoverable media error occurs during the restore operation, this parameter can be used to restart the operation.

This parameter is valid only when *NONSYS, *IBM, or *ALLUSR is specified for the restore operation.

Note:

In the recovery steps that follow, *NONSYS is specified on the SAVLIB parameter. If you are restoring IBM-supplied libraries or all user-created libraries and IBM-supplied libraries, specify *IBM or *ALLUSR instead.

The basic recovery steps for a restore operation are:

1. Look at the job log to determine the library where the previous restore library (RSTLIB SAVLIB(*NONSYS)) command failed. Find the last library restored, which is indicated by a successful restore completion message.
2. Load the first tape of the SAVLIB LIB(*NONSYS) media.
3. Issue the following command:

```
RSTLIB SAVLIB(*NONSYS) DEV(tape-name)
      ENDOPT(*LEAVE) STRLIB(library-name)
      OMITLIB(library-name)
```

where the *library-name* for the STRLIB and OMITLIB parameters is the library where the RSTLIB failed. This starts the restore operation on the library after the library where the restore operation failed.

4. When you are prompted, load the volume containing the starting library.
5. After the restore operation is complete, restore the library where the restore operation failed using the media from a previous save operation.

Note:

Consider removing the tape with the media error from the next save rotation cycle to avoid another tape error.

***FIRST:** The restore operation begins with the first library saved.

library-name: Specify the name of the library where the restore operation begins.

OMITLIB

Specifies a list of libraries to be excluded from the restore operation.

Note: This parameter is valid only when *NONSYS, *IBM, or *ALLUSR is specified on the SAVLIB parameter.

***NONE:** No libraries are excluded from the restore operation.

library-name: Specify the names of the libraries to be excluded from the restore operation.

SAVF Specifies the qualified name of the save file used to restore the save data.

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

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

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

library-name: Specify the name of the library to be searched.

save-file-name: Specify the name of the file from which to perform the restore operation.

MEDDFN

Specifies the qualified name of the media definition that identifies the devices and media used to restore the data. For information about creating and using a media definition, see the Setting up saves to multiple devices article in the Backup, Recovery, and Media Services topic in the Information Center.

Note: If a media definition is specified, the VOL, SEQNBR, SAVF, and OPTFILE parameters cannot be specified. The volume identifiers and sequence numbers are specified in the media definition.

The name of the media definition can be qualified by one of the following library values:

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

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

library-name: Specify the name of the library to be searched.

media-definition-name: Specify the name of the media definition.

OPTION

Specifies how to handle restoring each object.

***ALL:** All the objects in the saved library are restored to the library. Old objects on diskette, tape, or in a save file replace the current versions in the library on the system, and the objects not having a current version are added to the library on the system. Objects presently in the library but not on the media, remain in the library.

***NEW:** Only the objects in the saved library that do not exist in the current version of the library on the system are added to the library. Only objects not known to the library on the system are restored; known objects are not restored. This option restores objects that were deleted after they were saved or that are new to this library. If any saved objects have a version already in the library on the system, they are not restored, an informational message is sent for each object, and the restore operation continues.

***OLD:** Only the objects in the library having a saved version are restored; that is, the version of each object currently in the library is replaced by the saved version. Only objects known to the library are restored. If any saved objects are no longer part of the online version of the library, they are not added to the library and an informational message is sent for each object, but the restore operation continues.

***FREE:** The saved objects are restored only if they exist in the library on the system with their space freed. The saved version of each object is restored on the system in its previously freed space. This option restores objects that had their space freed when they were saved. If any saved objects are no longer part of the current version of the library, or if the space is not free for any object, the object is not restored and an informational message is sent for each object. The restore operation continues, and all of the freed objects are correctly restored.

MBROPT

Specifies, for database files currently on the system, which file members are restored.

Note:

If MBROPT(*MATCH) is used, the member list in the saved file must match, member for member, the current version in the system. All members are restored for files that do not exist. Before restoring a file, the system verifies whether the file and member creation dates of the known system objects match the creation dates of the objects being restored. If verification fails, the file is not restored.

***MATCH:** The saved members are restored if the lists of the members where they exist match, member for member, the lists of the current system version. *MATCH cannot be specified if *ALL is specified on the ALWOBJDIF parameter.

***ALL:** All members in the saved file are restored.

***NEW:** Only new members (members not known to the system) are restored.

***OLD:** Only members already known to the system are restored.

SAVDATE

Specifies the date the library was saved. If the most recently saved version is not restored, or if more than one saved version is on the volumes, enter the date that specifies which version of the library is restored. The date must be entered in the job date format. If separators (specified by the system value QDATSEP) are used, the value must be enclosed in apostrophes. If a volume identifier or VOL(*MOUNTED) is specified, but SAVDATE is not, the first version of the library found is restored. This parameter is valid only if a volume identifier is specified, VOL(*MOUNTED) is specified, or if a save file (SAVF parameter) is specified. Specify the date when the file was saved.

SAVTIME

Specifies the time when the library was saved, if the current version is not restored. 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.

If a volume identifier or *MOUNTED is specified on the VOL parameter, but this parameter is not, the first version of the library found on the volume is restored. This parameter is valid only if the SAVDATE parameter is also specified.

This parameter is valid only if SAVDATE is also specified.

ALWOBJDIF

Specifies whether certain differences encountered during a restore operation are allowed. The differences include:

- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.
- Primary Group: The primary group of an object on the system is different than the primary group of an object from the save operation.
- File creation date: The creation date of the database file on the system does not match the creation date of the file that was saved.
- File level identifier: The file and/or member level identifier of a physical file on the save media is different than the file and/or member level identifier of the physical file on the system.
- Member creation date: The creation date of the database file on the system does not match the creation date of the member that was saved.
- Validation value verification: The validation value created at the time an object was created does not match the validation value created during the restore operation of an object on a system with a QSECURITY level of 40 or higher.
- Authorization list linking: The object is being restored to a system different from the one on which it was saved.

Note:

The user of this parameter must have *ALLOBJ authority. If *ALL is specified, *MATCH cannot be specified on the MBROPT parameter.

***NONE:** None of the differences described above are allowed on the restore operation. For validation value verification failure cases, the object is restored but ownership is transferred to QDFTOWN and all authorities are revoked. For authorization list cases, the object is restored, but the object is not linked to the authorization list, and public authority is set to *EXCLUDE. For all other cases, a diagnostic message is sent for the object, and the object is not restored.

***ALL:** All of the differences listed above are allowed for the restore operation. An informational message is sent, except for validation value verification and authorization list cases, and then the object is restored. The following facts should be noted:

- If the owners of the object do not match, the object is restored, but the ownership and authorities it had before the restore operation are retained.
- If the primary groups of the object do not match, the object is restored, but the primary group it had before the restore operation is retained.
- Either the file itself or members of the file are restored based on the following facts:

1. If there is a file-level mismatch and ALWOBJDIF(*ALL) has been specified, the existing version of the file is renamed and the saved version of the file is restored.
 2. If there is a member-level mismatch, the existing version of the member is renamed and the saved version of the member is restored.
- Validation value verification:
 1. For programs with no validation value stored on the media (in reference to program data saved on an OS/400 system prior to version 1 release 3.0), the program is restored and the system does not attempt to recreate it. Nothing is logged in the security journal and no informational messages are sent.
 2. For programs that do have a validation value stored on the media and the system fails to recreate the program, the original version of the program is restored and an entry is logged in the security journal and job log.
 - The informational message causes a diagnostic message to be sent indicating that security or integrity changes occurred during the restore operation. This results in sending an escape message for the restore operation.
 - If the user is restoring objects to a system different from the one on which they were saved and the objects are secured by an authorization list, specifying *ALL automatically links the objects to the authorization list again. If the authorization list does not exist on the new system, a message that includes the name of the missing list is issued and the public authority is set to *EXCLUDE.

***FILELVL:** The file and/or member level identifier of a physical file can be different. If a database physical file already exists on the system with a different file and/or member level identifier than the saved physical file, then an attempt will be made to restore the physical file data. The file and/or member level identifiers of the physical file on the system remain unchanged. Objects other than physical files will be restored as if ALWOBJDIF(*NONE) is specified.

FRCOBJCVN

Specifies whether to convert user objects to the format required for use in the current version of the operating system when the objects are restored.

Notes:

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

Element 1: Force Conversion

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

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



➤ Note:

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

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

Notes:

1. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

2.  If FRCOBJCVN(*YES *RQD) is specified, then the QFRCCVNRST system value must have a value of "0," "1," or "2,". FRCOBJCVN(*YES *RQD) will override a QFRCCVNRST value of "0," or "1,". If FRCOBJCVN(*YES *ALL) is specified, then QFRCCVNRST can have any valid value and FRCOBJCVN(*YES *ALL) overrides the QFRCCVNRST system value. 

Element 2: Objects to Convert

***RQD:** The objects are converted only if they require conversion to be used by the current operating system. If the objects are not observable, the objects are restored but are not converted.

***ALL:** All objects are converted regardless of their current format. Even if the objects are in the current format, they are converted again. However, if the objects are not observable, the objects are not restored.

RSTLIB

Specifies whether the library contents are restored to the same library in which they were saved, or to a different library. If a different library is specified, the user cannot specify *NONSYS, *ALLUSR, or *IBM on the SAVLIB parameter.

***SAVLIB:** The library contents are restored to the same library or libraries in which they were saved.

library-name: Specify the name of the library where the saved library contents are being restored. If *NONSYS, *ALLUSR, or *IBM is specified on the SAVLIB parameter, a library name cannot be specified on this parameter.


Note:

If an SQL database is restored to a library other than the one in which it was saved, the journals are not restored.


RSTASPDEV

Specifies the name of the auxiliary storage pool (ASP) device to which the data is restored.

***SAVASPDEV:** The data is restored to the same auxiliary storage pool from which it was saved.

auxiliary-storage-pool-device-name: Specify the name of the independent ASP to which you want to restore the data. 

RSTASP

 Specifies whether the libraries and objects are restored to the same auxiliary storage pool (ASP) from which they were saved or to the system ASP (ASP number 1) or to a basic user ASP (ASP numbers 2-32).

A basic user ASP can contain either: 

- Libraries and all objects in the libraries.
- Journals, journal receivers, and save files for libraries in the system ASP.

This parameter applies to all libraries that are specified. More information about user ASPs is in the Backup, Recovery, and Availability topic in the Information Center.

Attention:

System or product libraries (libraries that begin with a Q or #) must not be created in or restored to a user ASP. Doing so can cause unpredictable results.

***SAVASP:** The objects are restored to the same auxiliary storage pools from which they were saved.

auxiliary-storage-pool-number: Specify the ASP number. When the specified ASP is 1, the objects are restored to the system ASP, otherwise the objects are restored to the basic user ASP specified. > To restore libraries and objects to an independent ASP, use the RSTASPDEV parameter. <<

OUTPUT

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

***NONE:** No output listing is created.

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

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

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

OUTFILE

Specifies the qualified name of the database file to which the information is directed when *OUTFILE is specified on the OUTPUT parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QASRRSTO in QSYS with the format name QSRRST as a model.

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

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

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

library-name: Specify the name of the library to be searched.

database-file-name: Specify the name of the database file to which output from the command is directed. If this file does not exist, it is created in the specified library. More information on defining the format of database files as output files is in the Database Programming topic in the Information Center.

OUTMBR

Specifies the name of the database file member to which the output is directed. This parameter is valid only if OUTPUT(*OUTFILE) is specified.

Element 1: Member to Receive Output

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

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

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

Element 2: Operation to Perform on Member

***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.

INFTYPE

Specifies the type of information directed to the database file. This parameter is valid only if OUTPUT(*OUTFILE) is specified.

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

***MBR:** The list contains an entry for each object or, for database files, each member requested to be restored.

OPTFILE

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume. For more information on specifying path names, refer to path names.

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

'optical-directory-path-name/*': The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

'optical-file-path-name': Specify the path name of the optical file.

Examples for RSTLIB

Example 1: Restoring New Objects

```
RSTLIB SAVLIB(JOE) DEV(TAP01) OPTION(*NEW)
```

This command restores the saved version of library JOE from tape device TAP01. The only objects that are restored in the library are new objects (ones that were in the library when they were saved and later deleted).

Example 2: Printing Output

```
RSTLIB SAVLIB(*NONSYS) DEV(TAP01) OUTPUT(*PRINT)
```

This command restores all the saved user-created libraries, the QGPL library, and the licensed program libraries (such as QRPGL and QIDU) to the system from tape. The contents of the libraries are restored exactly as they were saved. New objects (on tape) are added to the system; old objects in the system are overlaid by the version of the old objects on tape. Because OUTPUT(*PRINT) is specified, a printout of all objects (restored and not restored) for each library, is sent to the printer with the job's spooled output. Each library after the first library starts on a new page. After each library, a completion message states how many objects were restored and how many were not restored. At the end of a list, a final completion message states how many libraries were restored and how many were not restored.

Example 3: Specifying Where the Restore Operation Begins

```
RSTLIB SAVLIB(*NONSYS) DEV(TAP01) STRLIB(MIKESLIB)
```

This command restores the saved nonsystem libraries beginning with library MIKESLIB from the tape device, TAP01. The nonsystem libraries are saved in alphabetic order. Therefore, all libraries beginning with and following the name MIKESLIB are restored. The *first* tape of the nonsystem save must be loaded. An inquiry message instructs the user to load the tape containing MIKESLIB. If necessary, the same message is sent until the tape containing MIKESLIB is found.

Example 4: Restoring a Version From a Specific Date and Time

```
RSTLIB SAVLIB(PAYROLL) DEV(TAP01) SAVDATE(060193)
SAVTIME(103214) RSTLIB(OLDPAY) VOL(PAY)
```

This command restores the version of the PAYROLL library from the device TAP01, whose volume identifier is PAY. The version to be restored was saved at 10:32:14 on the date 06/01/93. All of the objects in the saved PAYROLL library are restored to the library OLDPAY. All new files are restored. Old files are restored only if the member lists of the files on the tape match the member lists of the files on the system.

Example 5: Restoring From Multiple Tape Volumes

```
RSTLIB SAVLIB(QGPL) DEV(TAP01) VOL(QGPL QGPL)
```

This command restores the QGPL library from two tape volumes both named QGPL. Even though the volume identifiers are the same, they must both be specified.

Example 6: Restoring From Multiple Tape Devices Serially

```
RSTLIB SAVLIB(USRLIB) DEV(TAP01 TAP02 TAP03)
VOL(USRA USRB USRC USRD) ENDOPT(*UNLOAD)
```

This command restores library USRLIB from four volumes on three tape devices. Volume USRA is put on tape device TAP01, volume USRB on TAP02, volume USRC on TAP03, and volume USRD on TAP01. The operator removes volume USRA from TAP01, so that TAP01 can be used by volume USRD. If the tape volumes are put in the wrong order, an error message is sent to the system operator message queue.

Example 7: Restoring From Multiple Devices in Parallel

```
RSTLIB SAVLIB(USRLIB) DEV(*MEDDFN) MEDDFN(LIBA/MEDDFNA)
```

This command restores library USRLIB using the devices and volumes specified in the media definition named MEDDFNA in library LIBA. For information about creating and using a media definition, see the Backup, Recovery, and Availability topic in the Information Center.

Example 8: Restoring a Specific Version

```
RSTLIB SAVLIB(LIB1) DEV(TAP01) MBROPT(*ALL)
SAVDATE(082392) SAVTIME(123251)
RSTLIB(LIB2) OUTPUT(*PRINT)
```

This command restores the version of library LIB1 from the device TAP01. The version to be restored was saved at 12:32:51 on the date 08/23/92. All of the objects in the saved library LIB1 are restored to library LIB2. A list of restored objects and those not restored is given. All files and file members are restored.

Example 9: Restoring a Library From a Save File

```
RSTLIB SAVLIB(LIB1) DEV(*SAVF) SAVF(SAVF1)
```

This command restores library LIB1 from the save file SAVF1.

Example 10: Restoring to a Basic User ASP

```
RSTLIB SAVLIB(LIB1) DEV(*SAVF) SAVF(SAVF1)
RSTASP(2)
```

This command restores the library named LIB1 from the save file named SAVF1. The library and all objects in the saved version of LIB1 are restored to basic user ASP 2 unless:

- The library already exists in a different ASP.
- ASP 2 contains a journal, journal receiver, or SAVF which is part of a library in the system ASP.
- ASP 2 does not exist on the system.
- There are object types in the library which cannot be restored to user ASPs. These objects will not be restored.

» Example 11: Restoring to an Independent ASP

```
RSTLIB SAVLIB(LIB1) DEV(*SAVF) SAVF(SAVF1)
RSTASPDEV(DIVISION1)
```

This command restores the library named LIB1 from the save file named SAVF1. The library and all objects in the saved version of LIB1 are restored to the independent ASP named DIVISION1 unless:

- The library already exists in the system ASP, a basic user ASP, or a different independent ASP in the same linked ASP group as DIVISION1.
- ASP DIVISION1 is not available on the system.
- There are object types in the library which cannot be restored to independent ASPs. These objects will not be restored. <<

Error messages for RSTLIB

*ESCAPE Messages

CPFACDE

Unconverted M36 exists. Restore not allowed.

CPF370C

Not authorized to ALWOBJDIF parameter.

CPF3705

&2 &1 in &3 not journaled.

CPF3706

&2 &1 not restored to library &3.

CPF3707

Save file &1 in &2 contains no data.

CPF3709

Tape devices do not support same densities.

CPF372C

Library &1 not restored to ASP &2.

CPF3727

Duplicate device &1 specified on device name list.

CPF3728

Device &1 specified with other devices.

CPF373E

Library &1 not restored to ASP &2.

CPF3730

Not authorized to &2 &1 in library &3.

CPF3731

Cannot use &2 &1 in library &3.

CPF3732

&2 &1 status changed during restore operation.

CPF3733

&2 &1 in &3 previously damaged.

CPF3738

Device &1 used for save or restore is damaged.

CPF3739

Database file &1 member in &3 damaged.

CPF3740
Object &1 type &2 in &3 not found.

CPF3743
File cannot be restored, displayed, or listed.

CPF3748
Object information for library &1 damaged.

CPF375F
File not selected. Cannot restore from save type file &1.

CPF3752
No record of save operation exists for library &1.

CPF3757
&2 &1 not restored to &3.

CPF3758
&2 &1 not restored to &3.

CPF376B
File &1 not found.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF3769
File found on media not save/restore file.

CPF3770
No objects saved or restored for library &1.

CPF3773
&1 objects restored. &2 not restored to &4.

CPF3779
&1 libraries restored; &4 partially restored; &2 not restored.

CPF378B
Library &1 not created.

CPF3780
Specified file for library &1 not found.

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

CPF3783
Cannot determine VOL(*SAVVOL) location. No objects restored.

CPF3784
Restore device specified in the DEV parameter does not match VOL(*SAVVOL) device.

CPF3785
Not all subsystems ended.

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

CPF3793
Machine storage limit reached.

- CPF3794**
Save or restore operation ended unsuccessfully.
- CPF3796**
Storage limit exceeded for user profile &4.
- CPF380E**
User profile &1 not restored. Not able to create UID or GID.
- CPF3805**
Objects from save file &1 in &2 not restored.
- CPF3807**
Data decompression error for save file &1 in &2.
- CPF3812**
Save file &1 in &2 in use.
- CPF3818**
Starting library &1 not found.
- CPF384F**
&2 &1 not restored to library &3.
- CPF3894**
Cancel reply received for message &1.
- CPF5729**
Not able to allocate object &1.
- CPF9809**
Library &1 cannot be accessed.
- CPF9812**
File &1 in library &2 not found.
- » **CPF9814**
Device &1 not found. «
- CPF9820**
Not authorized to use library &1.
- CPF9822**
Not authorized to file &1 in library &2.
- CPF9825**
Not authorized to device &1.
- CPF9829**
Auxiliary storage pool &1 not found.
- » **CPFB8ED**
Device description &1 not correct for operation. «
- OPT1498**
Volume name list exhausted on device &1.
- OPT1502**
Attempted to process past the end of a multi-volume set.
- OPT1605**
Media or device error occurred.

***STATUS Messages**

CPF3770

No objects saved or restored for library &1.

CPF3773

&1 objects restored. &2 not restored to &4.

RSTLIBBRM (Restore Library Using BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.

RSTLIBBRM Command syntax diagram

Purpose

The Restore Library using BRM (RSTLIBBRM) command restores a library that is in the BRMS media information. Any library that was saved by the Save Library using BRM (SAVLIBBRM) command or was in a control group that was saved can be restored by the RSTLIBBRM command. The RSTLIBBRM command restores the whole library, which includes the library description, object descriptions and contents of the objects in the library.

➤ Restrictions:

- You must have authority to the RSTOBJ command to use this command.
- You must have *USE authority to any auxiliary storage pool device specified for the RSTASP parameter.



Notes:

1. Do not precede an entry with an asterisk unless that entry is a “special value” that is shown (on the display itself or in the help information) with an asterisk.
2. Using the RSTLIBBRM command allows you to restore selected copies of a library from the BRMS media content information. For instance, a recovery request for a copy of a library will restore the full backup of the library that corresponds to the copy requested in the COPY parameter, plus the last incremental backup of that library preceding the next full backup of the library, plus any object level saves that are between the two full saves. This assures that the copy that you requested has all changes applied prior to the next full save.
3. You can restore data from a TSM server device by using this command. You can only specify one TSM device or *MEDCLS, which must select a TSM device. The TSM device selected can either be *APPC, which supports the SNA protocol, or *NET, which supports the TCP/IP protocol.

Note: This command should not be used by control group *EXIT item processing as results will be unpredictable.

Example for RSTLIBBRM

Example 1: Restoring a Library using BRMS

```
RSTLIBBRM SAVLIB(MBALIB) DEV(*MEDCLS)
COPY(2) RSTASP(3)
```

This command allows you to select and restore libraries or BRMS media information that were saved using a control group or a BRMS save command. In the example, the restore operation is for a library called MBALIB. Any device that supports the media class assigned to the media containing MBALIB can be used in the restore operation. The second copy of the library in the media information is restored to auxiliary storage pool number 3.

Error messages for RSTLIBBRM

None

RSTLICPGM (Restore Licensed Program) Command Description

RSTLICPGM Command syntax diagram

Purpose


The Restore Licensed Program (RSTLICPGM) command loads or restores a licensed program, either for initial installation or new-release installation.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. To use this command, the user must have *SECADM and *ALLOBJ authority.
3. If this command is used to restore a program in the licensed program, the copy of the program currently in the system should not be running while the program is being restored. If this occurs, the processing program is ended abnormally.
4. If other objects of the licensed program are in use, they are not restored.
5. With the exception of overrides for the restore operation printing OUTPUT(*PRINT), this command ignores all file overrides currently in effect for the job.
6. Some licensed programs can only be restored if the user is enrolled in the system distribution directory. See the publication for each licensed program for a description of this restriction.
7. This command does not restore code and language objects for the base OS/400 system.
8. This command does not support the use of other ASPs (auxiliary storage pools). All objects supplied by a licensed program must remain in the system ASP.

Required Parameters

LICPGM

Specifies the 7-character identifier of the licensed program being restored. A list of IBM-supplied licensed programs is in the Software Installation  book.

DEV Specifies the name of the device from which the licensed program objects are restored. The device name must be known by a device description on the system. If multiple devices are specified, they must have compatible media formats.

Up to four device names can be specified. Use the Work with Device Descriptions (WRKDEVD) command to display the names of the tape devices available on this system. Only one save file name can be specified.

***SAVF:** The restore operation is done using the save file name specified on the SAVF parameter.

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

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

tape-device-name: Specify the names of one or more tape devices used for the restore operation. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Optional Parameters

VOL Specifies the volume identifiers of the media volumes from which the licensed program is restored. The volumes must be installed on the device in the same order they were in when the licensed program was saved. The volume that contains the beginning of the file to be restored should be mounted on the device. A maximum of 75 entries can be specified.

***MOUNTED:** The licensed program is restored from the volumes that are mounted on the device specified on the DEV parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

volume-identifier: Specify the identifiers of one or more volumes in the order they are mounted on the device and used to restore the licensed program.

OPTION

Specifies which licensed program option to restore.

***BASE:** Only the base part of the licensed program is restored.

number-of-licensed-program-option: Specify the number associated with the optional part of the listed licensed program being restored.

RSTOBJ

Specifies the type of licensed program objects being restored.

***ALL:** All objects of the licensed program are restored. This includes both the program objects and the language objects.

If a tape device is specified on the DEV parameter, then the RSTOBJ(*ALL) value is used when the saving of the licensed program has been done with the SAVLICPGM command such that the language objects immediately follow the program objects on the tape media. If the language objects (*LNG) and programming objects (*PGM) are not in consecutive order on the distribution tape, *ALL cannot be used in most cases. Instead, the program and language objects must be restored separately. The DSPTAP command can be used to determine the order of the objects on the tape. An example of how to restore language and program objects separately is in the "Examples" section at the end of this command.

If *SAVF is specified on the DEV parameter, then the RSTOBJ(*ALL) value can be used when the saving of the licensed program has been done with the SAVLICPGM command using OBJTYPE(*ALL).

***PGM:** Only the program objects for the licensed program are restored. This value is used when restoring the program objects from the distribution media where the program objects and the selected language objects are not on the same distribution media or are not in consecutive order.


***LNG:** The objects associated with the language identified by the language feature indicated on the LNG parameter are restored.

LNG Specifies the language to be used for restoring the licensed program. If the language feature of the licensed program on the save media matches the system language feature, the language objects are restored to the licensed program's libraries. If the language features do not match, the language objects are restored into the multilingual library for that language feature.

***PRIMARY:** The language feature of the operating system is restored for the specified product.

Note: Use the GO LICPGM function with option 20 to display the primary language of the operating system.

***SAVVOL:** The language file on the mounted volume is to be restored for the licensed program specified by the user. This option is not valid with DEV(*SAVF).

feature-code: Specify the language feature identifier for the language file that is to be restored for the licensed program specified by the user. More information on feature identifications and a list of the IBM-supplied feature codes is in the Software Installation  book.

SEQNBR

Specifies which sequence number to use for the restore operation.

***SEARCH:** The volume on the device is searched for a data file for the licensed program; when a match is found, the objects are restored.

file-sequence-number: Specify the sequence number of the file. Valid values range from 1 through 16777215.

ENDOPT

Specifies the operation that is automatically performed on the tape or optical volume after the restore operation ends. If one or more volumes of tape is involved, this parameter applies only to the last volume.

Note:

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

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

***LEAVE:** The tape is not rewound. Another restore operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the restore operation has ended. Some optical devices will eject the volume after the operation ends.

SAVF Specifies the qualified name of the save file containing the product.

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

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

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

library-name: Specify the name of the library to be searched.

save-file-name: Specify the name of the save file.

OUTPUT

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

***NONE:** No output is created.

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

RLS Specifies the version, release, and modification level of the product being restored.

***FIRST:** The first version, release, and modification level found on the distribution media is restored.

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

REPLACERLS

Specifies the version, release, and modification level of the product being replaced.

***ONLY:** Replace only the version, release, and modification level of the product currently installed.

***NO:** The product currently installed on the system is not replaced. The product being restored must be a different release than the product currently installed. If the product being restored exists in the same libraries as the installed product, then an override parameter must be specified indicating to which libraries the product is restored.

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

LIB Specifies the libraries into which the product is being restored. This function is not supported by all products.

***SAME:** The product is restored into the library specified by the vendor.

library-name: Specify the name of the library into which the product is being restored.

LNLIB

Specifies the secondary language library into which the product is being restored. This function is not supported by all products.

***SAME:** The product is restored into the secondary language library specified by the vendor.

library-name: Specify the name of the secondary language library into which the product is restored.

FLR Specifies the name of the root folder into which the product is being restored. This function is not supported by all products.

***SAME:** The root folder specified by the vendor is used.

folder-name: Specify the name of the root folder.

CODHOMEDIR

Specifies the directories into which the code part of the product is being restored. This function is not supported by all products.

Note:

This parameter is mutually exclusive with the FLR parameter.

***SAME:** The code part of the product is restored into the directories specified when packaged or already installed. *SAME may be specified as the only parameter value or within a list of directories. If used within a list, *SAME specifies that a particular directory is unchanged, though other directories may be different than when the product was packaged or previously installed.

***PROMPT:** The code directories to be used are displayed. If the product is not currently installed, the directory names can be changed.

path-name: Specify the home path directory name into which the code part of the product is being restored. Up to 300 directories may be specified. For directory name entries which are unchanged, *SAME can be specified for the path name.

LNGHOMEDIR

Specifies the directories into which the language part of the product is being restored. This function is not supported by all products.

Note: This parameter is mutually exclusive with the FLR parameter.

***SAME:** The language part of the product is restored into the directories specified when packaged or already installed. *SAME may be specified as the only parameter value or within a list of directories. If used within a list, *SAME specifies that a particular directory is unchanged, though other directories may be different than when the product was packaged or previously installed.

***PROMPT:** The language directories to be used are displayed. If the product is not currently installed, the directory names can be changed.

path-name: Specify the home path directory name into which the language part of the product is being restored. Up to 300 directories may be specified. For directory name entries which are unchanged, *SAME can be specified for the path name.

FRCOBJCVN

Specifies whether to convert user objects to the format required for use in the current version of the operating system when the objects are restored.

Notes:

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

Element 1: Force Conversion

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

Note: If this value is specified or defaulted and the system value QFRCCVNRST has a value of "1," the restore operation proceeds as if FRCOBJCVN(*YES *RQD) is specified.

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

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

Note: Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

Element 2: Objects to Convert

***RQD:** The objects are converted only if they require conversion to be used by the current operating system. If the objects are not observable, the objects are restored but are not converted.

***ALL:** All objects are converted regardless of their current format. Even if the objects are in the current format, they are converted again. However, if the objects are not observable, the objects are not restored.

Examples for RSTLICPGM

Example 1: Restoring Program Using Defaults

```
RSTLICPGM LICPGM(5716CD1) DEV(TAP01)
```

This command restores the CoOperative Development Environment/400 (CODE/400) licensed program to the system. The tape containing the licensed program objects must be put on the TAP01 tape drive. Because no other parameters are specified, the defaults are used for the command.

Example 2: Restoring a Third Version of a Product

```
RSTLICPGM LICPGM(1MYPROD) OPTION(*BASE) DEV(TAP01)
  RLS(V3R6M0) REPLACERLS(*NO) LIB(A B C)
```

This command restores the base part of the V3R6M0 1MYPROD product to the system if the base of the V3R6M0 1MYPROD product is not currently installed on the system.

Example 3: Restoring One Version of a Product Over Another Version

```
RSTLICPGM LICPGM(2MYPROD) OPTION(*BASE) DEV(TAP01)
  RLS(*FIRST) REPLACERLS(*ONLY)
```

This command restores the first version release modification level of the base part of the 2MYPROD product that is found on the tape in the TAP01 drive. It does not matter if the version release modification level of the base of the product on the tape matches the version release modification level of the base of the product on the system.

Example 4: Restoring Product From Save File

```
RSTLICPGM LICPGM(5716CD1) DEV(*SAVF)
  SAVF(MYLIB/MYSAVF)
```

This command restores the CODE/400 licensed program to the system from the save file MYSAVF in MYLIB. Because no other parameters are specified, the defaults are used for the command.

Example 5: Restoring a Third Version of a Product From a Save File

```
RSTLICPGM LICPGM(1MYPROD) OPTION(*BASE) DEV(*SAVF)
  RLS(V3R6M0) REPLACERLS(*NO) LIB(A B C)
  SAVF(MYLIB/MYSAVF)
```

This command restores the base part of the V3R6M0 1MYPROD product to the system from save file MYSAVF in MYLIB if the base of the V3R6M0 1MYPROD product is not currently installed on the system.

Error messages for RSTLICPGM

*ESCAPE Messages

CPF3D94

No product found in save file.

CPF3D96

Objects for product &1 option &2 release &4 not restored.

CPF37A2

Licensed program &1 not valid.

CPF3728

Device &1 specified with other devices.

CPF3733

&2 &1 in &3 previously damaged.

CPF3739

Database file &1 member in &3 damaged.

CPF3820

&4 objects for &1 option &2 not restored.

CPF3880

No language objects exist.

CPF3884

Licensed program &1 option &2 not processed.

CPI36C9

Error occurred while removing PTFs.

RSTOBJ (Restore Object) Command Description

RSTOBJ Command syntax diagram

Purpose

The Restore Object (RSTOBJ) command restores to the system a single object, or a group of objects, in a single library, that were saved on diskette, tape, optical volume, or in a save file by using a single command. The types of objects that can be restored by this command are listed in the OBJTYPE parameter. They can be saved either as separate objects or as part of the library save operation. The RSTOBJ command restores the object description and contents of each object specified in the command.

For job queues, output queues, user-defined message queues, logical files, and data queues, only the object descriptions are restored; the contents of those objects are not restored. However, logical file access paths can be saved by a save command with ACCPTH(*YES) specified; if this is done, the access paths can be restored. More information on restoring access paths is in the Database Programming topic in the Information Center.

The command can be used to restore the objects even if the object storage was freed by the Save Object (SAVOBJ) command or Save Library (SAVLIB) command, or if the objects were deleted by the associated delete command for that object type. If the storage was not freed, each object is restored in the same area of storage that it previously occupied. If the version of the object being restored is larger than the version in the system (for example, data records that were deleted from the system still exist in the saved version of a file), the additional storage needed for the object is acquired. If the saved version of the object is smaller (for example, data records that are added to the system), the space that was acquired for the object remains assigned to that object and available for use by that object.

If logical file access paths were saved (ACCPTH(*YES) was specified when the objects were saved), the access paths are restored if (1) all based-on physical files are also being restored by the same restore command, (2) the logical file is also being restored by the same restore command, or the logical file already exists on the system (the same file exists, not a re-created version), and (3) MAINT(*IMMED or *DLY) is in effect for the logical file if it still exists on the system.

If the storage was freed, the system finds the storage space needed to store the contents (only the data portion) of each file, module, program, service program, journal receiver, and Structured Query Language

(SQL) package. If the objects do not exist on the system because they were deleted or they are being restored in a different system, the system must find the storage to store everything (the description and the data portion) about each unknown object.

The user profile of the system default owner (QDFTOWN) becomes the default owner of objects restored in the system whose owner is not known to the system.

If an object already exists in the library to which it is restored, the public and private authorities of the existing object are kept. If the object does not exist in the library, all public authorities are restored, but any private authorities must be granted again. For an existing output queue object that is actively spooling during the restore operation, or a data queue that already exists in the library, none of the attributes are restored, and a diagnostic message is issued.

If an object is being restored over an existing object on the system, the object auditing value of the existing object is kept. If the object is being restored as new to the system, the object auditing value is restored from the media.

Attention:

Do not use the Restore Object (RSTOBJ) command to restore licensed programs into library QSYS. Unpredictable results can occur.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The user must have use authority for the Create Save File (CRTSAVF) command when restoring a save file that does not currently exist on the system.
3. When saving or restoring to an existing database file using the OUTFILE parameter, the user must have execute authority to the output file library.
4. With the exception of overrides for the restore listing file and database files specified on the OUTFILE parameter, this command ignores all file overrides that are currently in effect for the job.
5. To use this command, the user must have the special authority *SAVSYS, specified in the user profile (SPCAUT parameter), or the following:
 - Add and execute authority for the specified library and save file library.
 - Object existence authority for, or be the owner of, each object specified if the object already exists in the library on the system. Object existence authority and use authority are required for message queue objects. If the object does not already exist in the library on the system, the user must have add authority for the user profiles that own each object being restored.
 - If VOL(*SAVVOL) is specified, use authority to the saved-from library.
6. The RSTOBJ command does not restore the library's data dictionary or its associated database files. To do this, the RSTLIB command must be used.
7. The user must have use authority for the save file to restore from the save file. In addition, the user must have use authority for the device description when restoring from tape, diskette, or optical device.
8. When using a media definition, the user must have *USE authority to the media definition and *EXECUTE authority to the media definition library.
9. If this command is used to restore a program, the copy of that program that is currently in the system must not be running while the program is being restored. If this occurs, the running program will not be restored.
10. Objects saved by separate commands must also be restored by separate commands. If a single command is used, some of the objects are not restored.

11. If the user domain object user space (*USRSPC), user index (*USRIDX), or user queue (*USRQ) is restored to a library that is not permitted in the system value QALWUSRDMN (allow user domain objects in libraries), the object is converted to the system domain.

Required Parameters

OBJ Specifies the names of one or more objects to restore. The objects specified are restored from the first file that meets the search criteria and contains any of the objects. If a tape or diskette contains multiple files for the same library, specify the tape file sequence number or the diskette file label to ensure that all of the correct objects are restored.

If the file does not contain all of the objects specified, diagnostic messages are issued for the objects not found. The completion message contains a count of objects restored and objects not restored.

If the OBJTYPE parameter is not specified when the command is run, all of the object types listed in the description of the OBJTYPE parameter are restored, if they are in the specified library on the media or in the save file and have the specified names.

***ALL:** All the objects saved from the specified library are restored, depending on the values specified for the OBJTYPE and OPTION parameters.

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

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

SAVLIB

Specifies the name of the library from which the objects were saved. If RSTLIB is not specified, this is also the name of the library to which the objects are restored.

DEV Specifies the name of the device used to restore the objects. The device name must already be known on the system by a device description.

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

***MEDDFN:** The restore operation is done using the devices and media identified in the media definition specified by the Media Definition (MEDDFN) parameter.

diskette-device-name: Specify the name of the diskette device used to restore objects.

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

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

tape-device-name: Specify the names of one or more tape devices used for the restore operation. If more than one tape device is specified, they are used serially in the order in which they are specified. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Optional Parameters

OBJTYPE

Specifies the types of OS/400 system objects that are restored. More information on this parameter is in commonly used parameters.

***ALL:** All object types that are specified by name and that were saved from the specified library are restored. If *ALL is also specified for the OBJ parameter, then all objects saved for that library are restored.

object-type: Specify the value for each of the types of objects that are restored (such as command (*CMD), file (*FILE), or program (*PGM)). This value can be repeated; the maximum repetitions varies depending on the environment.

The object types that can be restored also can be saved and restored by the Save Library (SAVLIB), Save Change Object (SAVCHGOBJ), Save Object (SAVOBJ), Restore Library (RSTLIB), and Restore Licensed Program (RSTLICPGM) commands. Data dictionaries (*DTADCT) are only restored with the RSTLIB command. Other object types can be saved only by the Save System (SAVSYS) command and restored by the OS/400 system restore operation, or saved by the Save Document Library Object (SAVDLO) command and restored by the Restore Document Library Object (RSTDLO) command.

Note: *CFGL object types can be restored using the RSTOBJ command only from objects saved on releases prior to V2R2M0. *CFGL object types saved on V2R2M0 and newer releases are restored using the Restore Configuration (RSTCFG) command.

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

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

***MOUNTED:** The objects are restored from the volumes placed in the device specified on the DEV parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

***SAVVOL:** The system, using the save/restore history information, determines which volumes contain the most recently saved version of the objects.

Note: If this is a restore from an optical device, only the first 6 characters of the volume name are available. If the volume name of the optical media exceeds 6 characters, the volume may not be found. You should specify the complete volume or file name on the command instead of using VOL(*SAVVOL) when the name of the optical media exceeds 6 characters.

When *SAVVOL is specified, the following operational characteristics and restrictions apply:

- If the characteristics of the device specified in the DEV parameter do not match the device and location of the most recently saved version of the library, an error message is sent to the *user*, and the function is ended.

- If the wrong volume is placed in a device in the location specified by the command, a message is sent to the *system operator* that identifies the first volume that must be placed in the device before the objects can be restored.
- When OBJ(*ALL) and OBJTYPE(*ALL) is specified, the objects are considered in the order they would appear in a display produced by the Display Library (DSPLIB) command. The object names and types specified in the RSTOBJ command are used to determine which file of saved objects is used in the restore operation. One file is produced for each SAVLIB or SAVOBJ command run. The file chosen is the one in which the first considered object was last saved. Objects that were not saved in the file chosen to be processed, or that were more recently saved, are not restored; an error message is sent to the user for each unrestored object.
- If *SAVVOL is specified, the SAVDATE and SAVTIME parameters cannot be specified.
- If *SAVVOL is specified and the RSTLIB value is equal to the SAVLIB value, OPTION(*NEW) cannot be specified.
- If *SAVVOL is specified, SEQNBR(*SEARCH) and LABEL(*SAVLIB) must be specified.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to restore the objects.

SEQNBR

Specifies, only when tape is used, the sequence number used for the restore operation.

***SEARCH:** The volume on a tape device is searched for a data file with an identifier that is a match for the SAVLIB parameter value; when a match is found, the object is restored. If the last operation on the device specifies ENDOPT(*LEAVE) (that is, the tape is positioned at the location where the last operation ended), the file search starts with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last operation (or if the tape was manually rewound after an ENDOPT(*LEAVE) operation), the search starts with the first data file on the volume.

sequence-number: Specify the sequence number of the file. Valid values range from 1 through 16777215.

LABEL

Specifies the name that identifies the data file on the tape or diskette used for the restore operation. This label must have been specified on the save command.

***SAVLIB:** The file label is the name of the library specified on the SAVLIB parameter.

data-file-identifier: Specify the data file identifier (up to 17 characters) of the data file used for the restore operation. This option is valid only for a single library restore.

ENDOPT

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

Note:

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

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

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

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

SAVF Specifies the qualified name of the save file that contains the save data for the objects that are restored.

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

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

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

library-name: Specify the name of the library to be searched.

save-file-name: Specify the name of the save file from which to perform the restore.

MEDDFN

Specifies the qualified name of the media definition that identifies the devices and media used to restore the data. For information about creating and using a media definition, see the Setting up saves to multiple devices article in the Backup, Recovery, and Media Services topic in the Information Center.

Note:

If a media definition is specified, the VOL, SEQNBR, SAVF, and OPTFILE parameters cannot be specified. The volume identifiers and sequence numbers are specified in the media definition.

The name of the media definition can be qualified by one of the following library values:

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

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

library-name: Specify the name of the library to be searched.

media-definition-name: Specify the name of the media definition.

OPTION

Specifies how to handle restoring each object.

***ALL:** All the objects in the saved library are restored to the library. All objects in the saved library are restored to the library on the system. Objects in the saved library replace the objects in the library on the system. Saved objects not on the system are added to the library on the system. Objects in the system library but not in the saved library remain in the library.

***NEW:** Only the objects in the saved library that do not exist in the current version of the library in the system are added to the library. Only objects not known to the library in the system are restored; known objects are not restored. This option adds objects that were deleted after they

were saved or that are new to this library. If any saved objects have a current version in the library in the system, they are not restored; an informational message is sent for each object, but the restore operation continues.

***OLD:** Only objects in the library that have a saved version are restored; the current version of each object is replaced by the saved version. Only objects known to the library are restored. If any saved objects are no longer part of the current version of the library, they are not added to the library; an informational message is sent for each object, but the restore operation continues.

***FREE:** The saved objects are restored only if they exist in the library in the system with their space freed. The saved version of each object is restored in the system in its previously freed space. This option restores objects that had their space freed when they were saved. If any saved objects are no longer part of the current version of the library, or if the space is not free for any object, the object is not restored and an informational message is sent for each object. The restore operation continues and all of the freed objects are correctly restored.

FILEMBR

Specifies the database file members to restore. This parameter is made up of two parts: the file name and the member name. This parameter cannot be specified if MBROPT(*MATCH) is specified.

Notes:

1. Each database file specified on the FILEMBR parameter must also be specified on the OBJ parameter by either its complete name, a generic name, or *ALL.
2. The OBJTYPE parameter value must be *ALL, or it must include the *FILE value.
3. Generic names are not valid for the database file name, but are allowed for the member name.
4. Duplicate file names are not allowed.

Element 1: Database File Names

***ALL:** The member list following this value applies to all files specified on the OBJ parameter.

database-file-name: Specify the name of the database file that contains the members to be restored. Up to 50 of the file/member combinations can be specified for a single command.

Element 2: Member Names

***ALL:** All members are restored from the specified file.

***NONE:** No members are restored from the specified file. Only the file description is restored.

generic-member-name:* Specify the generic name of the members to be restored from the specified file. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. For more information on the use of generic functions, refer to generic object names.

member-name: Specify the name of the member to be restored from the specified file.

Notes:

1. If the nongeneric member names are specified, all specified members must exist in the file in order for any part of the file to be restored.
2. If generic member names are specified, the file is not restored if it does not contain any members that match the generic names. For example, if the user specifies PAY* as a generic

member name, and the system is unable to find any members whose names start with PAY, the file is not restored. If no files specified by the FILEMBR parameter are restored because no members with the specific generic name can be found, a diagnostic message is sent, the restore operation ends, and an escape message is sent specifying the number of files not restored. If at least one of the files processed for the FILEMBR parameter is restored, the diagnostic message is not sent, and the number of files not restored is shown on the final completion message.

3. The members are first checked by using the MBROPT parameter. If a match is found, the members are checked against the list of the FILEMBR parameter.

MBROPT

Specifies, for database files currently on the system, which file members are restored.

If MBROPT(*MATCH) is used, the member list in the saved file must match, member for member, the current version on the system.

Note:

Before restoring a file, the system verifies that the file and member creation dates of the known system objects match the creation dates of the objects to be restored. If this check fails, the file is not restored.

***MATCH:** The saved members are restored if the lists of the files where they reside match, member for member, the lists of the current system version.

***ALL:** All members in the saved file are restored.

***NEW:** Only new members (members not existing on the system) are restored.

***OLD:** Only members already existing on the system are restored.

SAVDATE

Specifies the date the objects were saved. If the most recently saved version is not the one being restored, or if multiple saved versions reside on the media volume, specify the date that identifies which version of the objects to restore. The date must be specified in the job date format. If separators, specified by the system value QDATSEP, are used, the value must be enclosed in apostrophes. If a volume identifier or VOL(*MOUNTED) is specified, but SAVDATE is not, the restored version of the objects is the first version found on the volume. This parameter is valid only if a volume identifier or VOL(*MOUNTED) is specified, or if a save file (SAVF parameter) is specified. Specify the date when the objects were saved.

SAVTIME

Specifies the time that the objects were saved.

The time can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.

If a volume identifier or *MOUNTED is specified on the VOL parameter, but this parameter is not specified, the version of the objects to be restored is the first version found on the volume. This parameter is ignored when the SEQNBR parameter is specified.

This parameter is valid only if the SAVDATE parameter is also specified.

ALWOBJDIF

Specifies whether certain differences encountered during a restore operation are allowed. The differences include:

- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.
- The primary group of the object on the system is different than the primary group of the object from the save operation.
- File creation date: The file creation date of the database file on the system does not match the creation date of the file that was saved.
- File level identifier: The file and/or member identifier of a physical file on the save media is different than the file and/or member level identifier of the physical file on the system.
- Member creation date: The file creation date of the database file on the system does not match the creation date of the member that was saved. The member creation date on the system does not match the creation date from the save operation.
- Validation value verification: The validation value created at the time an object was created does not match the validation value created during the restore operation of an object on a system with a QSECURITY level of 40 or higher.
- Authorization list linking: The object is being restored to a system different from the one on which it was saved.

Notes:

1. The user of this parameter must have *ALLOBJ special authority.
2. If *ALL is specified, *MATCH cannot be specified on the MBROPT parameter.

***NONE:** None of the differences described above are allowed on the restore operation. For validation value verification failure cases, the object is restored but ownership is transferred to QDFTOWN and all authorities are revoked. For authorization list cases, the object is restored, but the object is not linked to the authorization list, and public authority is set to *EXCLUDE. For all other cases, a diagnostic message is sent for the object, and the object is not restored.

***ALL:** All of the differences listed above are allowed for the restore operation. An informational message is sent, except for valid value verification and authorization list linking cases, and the object is restored. The following facts must be noted:

- If the owners of the object do not match, the object is restored, but the ownership and authorities it had before the restore operation are retained.
- If the primary groups of the object do not match, the object is restored, but the primary group it had before the restore operation is retained.
- Either the file itself or members of the file are restored based on the following facts:
 1. If there is a file level mismatch and ALWOBJDIF(*ALL) has been specified, the existing version of the file is renamed and the saved version of the file is restored.
 2. If there is a member level mismatch, the existing version of the member is renamed and the saved version of the member is restored.
- Validation value verification:
 1. For programs with no validation value stored on the media (in reference to program data saved on an OS/400 system prior to version 1, release 3.0), the program is restored and the system does not attempt to recreate it. Nothing is logged in the security journal and no informational messages are sent.
 2. For programs that do have a validation value stored on the media and the system fails to recreate the program, the original version of the program is restored and an entry is logged in the security journal and job log.

- The informational message causes a diagnostic message to be sent indicating that security or integrity changes occurred during the restore operation. This results in sending an escape message for the restore operation.
- If the user is restoring objects to a system different from the one on which they were saved and the objects are secured by an authorization list, specifying *ALL automatically links the objects to the authorization list again. If the authorization list does not exist on the new system, a message that includes the name of the missing list is issued and the public authority is set to *EXCLUDE.

***FILELVL:** The file and/or member level identifier of a physical file can be different. If a database physical file already exists on the system with a different file and/or member level identifier than the saved physical file, then an attempt will be made to restore the physical file data. The file and/or member level identifiers of the physical file on the system remain unchanged. Objects other than physical files will be restored as if ALWOBJDIF(*NONE) is specified.

FRCOBJCVN

Specifies whether to convert user objects to the format required for use in the current version of the operating system when the objects are restored.

Notes:

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

Element 1: Force Conversion

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

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



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

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

Notes:

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

Element 2: Objects to Convert

***RQD:** The objects are converted only if they require conversion to be used by the current operating system. If the objects are not observable, the objects are restored but are not converted.

***ALL:** All objects are converted regardless of their current format. Even if the objects are in the current format, they are converted again. However, if the objects are not observable, the objects are not restored.

RSTLIB

Specifies whether the objects are restored to a different library or to the same library where they were saved.

***SAVLIB:** The objects are restored to the same library from which they were saved.

library-name: Specify the name of the library to which the saved objects are restored.

» RSTASPDEV

Specifies the name of the auxiliary storage pool (ASP) device to which the data is restored.

***SAVASPDEV:** The data is restored to the same auxiliary storage pool from which it was saved.

auxiliary-storage-pool-device-name: Specify the name of the independent ASP to which you want to restore the data. «

RSTASP

» Specifies whether the objects are restored to the same auxiliary storage pool (ASP) from which they were saved or to the system ASP (ASP number 1) or to a basic user ASP (ASP numbers 2-32). « This parameter affects only object types, journals, journal receivers, save files, and libraries. All other object types are restored to the ASP of the library unless the library is in a user ASP and the object type is not one of the valid user ASP object types. A list of the valid user ASP object types and additional information about user ASPs are in the Backup, Recovery, and Availability topic in the Information Center.

Journals, journal receivers, and save files can be restored to a user ASP with the library in the system ASP if the objects do not already exist in a different ASP, and if the ASP to which they are being restored contains no libraries.

User ASPs that contain one or more libraries cannot contain valid ASP object types, journals, journal receivers, and save files in libraries on the system ASP.

***SAVASP:** The objects are restored to the same auxiliary storage pools from which they were saved.

auxiliary-storage-pool-number: Specify the ASP number. When the specified ASP is 1, the objects are restored to the system ASP, otherwise the objects are restored to the basic user ASP specified. » To restore libraries and objects to an independent ASP, use the RSTASPDEV parameter. «

Note:

System libraries (for example, QSYS, QUSRSYS, and QDOC) must not be created in or restored to user ASPs. Doing so can cause unpredictable results.

OUTPUT

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

***NONE:** No output is created.

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

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

Note:

You must specify the database file name on the OUTFILE parameter when OUTPUT(*OUTFILE) is specified.

OUTFILE

Specifies the qualified name of the database file to which the information about the object is directed when *OUTFILE is specified on the OUTPUT parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QASRRSTO in QSYS with the format name QSRRST as a model.

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

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

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

library-name: Specify the name of the library to be searched.

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

OUTMBR

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

Element 1: Member to Receive Output

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

member-name: Specify the name of the file member that receives the output. If OUTMBR(*member-name*) is specified and the member does not exist, the system creates it. If the member exists, the user can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Operation to Perform on Member

***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.

INFTYPE

Specifies the type of information directed to the database file. This parameter is valid only when OUTPUT(*OUTFILE) is specified.

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

***MBR:** The list contains an entry for each object or, for database files, each member requested to be restored.

OPTFILE

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume. For more information on specifying path names, refer to path names.

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

*'optical-directory-path-name/**:* The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

'optical-file-path-name': Specify the path name of the optical file.

Examples for RSTOBJ

Example 1: Restoring Most Recently Saved Version

```
RSTOBJ OBJ(PAYROLL) SAVLIB(LIBX) DEV(TAP01)
      OBJTYPE(*PGM) VOL(*SAVVOL)
```

This command restores to LIBX the program named PAYROLL that was saved from LIBX. The tape drive named TAP01 is used to restore the most recently saved version of the program.

Example 2: Specifying Save Date and Time

```
RSTOBJ OBJ(PAY*) SAVLIB(LIBX) DEV(DKT01) VOL(ABCD)
      OPTION(*OLD) SAVDATE(102292)
      SAVTIME(143000) RSTLIB(LIBY)
```

All objects whose names start with PAY and that were saved from the library named LIBX on diskette volume ABCD at 14:30:00 on 10/22/92 are restored to LIBY. Volume ABCD must be put on the diskette device named DKT01. Because OPTION(*OLD) is specified, the only objects restored are those having the same object name and type both in LIBY on the system and in LIBX on diskette.

Example 3: Adding a New Program to the QGPL Library

```
RSTOBJ OBJ(NEWPROG) SAVLIB(QGPL) DEV(DKT01)
      OBJTYPE(*PGM) VOL(PGMS) OPTION(*NEW)
```

A new program named NEWPROG is added to the general purpose library, QGPL. It is restored from a volume labeled PGMS that is inserted in the diskette device named DKT01.

Example 4: Printing An Output List

```
RSTOBJ OBJ(*ALL) SAVLIB(LIB) DEV(TAP01) OBJTYPE(*PGM)
      VOL(SVOL) SEQNBR(2) SAVDATE(082392)
      SAVTIME(143000) OUTPUT(*PRINT)
```

All programs that were saved from the library named LIB that exist in the saved version on the tape volume named SVOL, sequence number 2, saved at 14:30:00 on 08/23/92, are restored to the library named LIB. An output list of all objects restored as well as objects not restored is provided.

Example 5: Restoring Journal Receivers

```
RSTOBJ OBJ(*ALL) SAVLIB(BACKUP) DEV(*SAVF)
      OBJTYPE(*JRNRCV) SAVF(SAVEJ) RSTASP(3)
```

All journal receivers that were saved from the library named BACKUP into the save file named SAVEJ are restored to the library named BACKUP. The journal receivers are restored to basic user ASP 3 (unless they already exist in the library named BACKUP and are in a different ASP or ASP 3 contains a library).

Error messages for RSTOBJ

*ESCAPE Messages

CPFACDB

M36 &1 in library &2 is active. Request not allowed.

CPFACDE

Unconverted M36 exists. Restore not allowed.

CPF370C

Not authorized to ALWOBJDIF parameter.

CPF3705
&2 &1 in &3 not journaled.

CPF3706
&2 &1 not restored to library &3.

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

CPF3709
Tape devices do not support same densities.

CPF3727
Duplicate device &1 specified on device name list.

CPF3728
Device &1 specified with other devices.

CPF3730
Not authorized to &2 &1 in library &3.

CPF3731
Cannot use &2 &1 in library &3.

CPF3733
&2 &1 in &3 previously damaged.

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

CPF3739
Database file &1 member in &3 damaged.

CPF374C
No objects restored to ASP &2.

CPF3743
File cannot be restored, displayed, or listed.

CPF3748
Object information for library &1 damaged.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF3769
File found on media not save/restore file.

CPF3770
No objects saved or restored for library &1.

CPF3773
&1 objects restored. &2 not restored to &4.

CPF3780
Specified file for library &1 not found.

CPF3781
Library &1 not found.

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

CPF3783
Cannot determine VOL(*SAVVOL) location. No objects restored.

CPF3784
Restore device specified in the DEV parameter does not match VOL(*SAVVOL) device.

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

CPF3793
Machine storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3796
Storage limit exceeded for user profile &4.

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

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

CPF3812
Save file &1 in &2 in use.

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

CPF3867
Contents of FILEMBR parameter not correct.

CPF3868
FILEMBR specified but OBJTYPE must be *ALL or *FILE.

CPF3871
No objects saved or restored; &3 objects not included.

CPF3872
Not all objects restored from library &3 to library &4.

CPF5729
Not able to allocate object &1.

CPF9809
Library &1 cannot be accessed.

CPF9812
File &1 in library &2 not found.

» CPF9814
Device &1 not found. «

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

CPF9829
Auxiliary storage pool &1 not found.

» CPFB8ED

Device description &1 not correct for operation. «

OPT1498

Volume name list exhausted on device &1.

OPT1502

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

OPT1605

Media or device error occurred.



RSTBRM (Restore Object Using BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.


RSTBRM Command syntax diagram

Purpose

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

For more information about the integrated files system commands, see the Integrated File System Introduction topic in the Information Center.

Restrictions:

1. For detailed restrictions on using this command to restore objects by using name patterns, in the root directory, to restore objects in libraries, or to restore document library objects, see the Backup and Recovery  book.
2. The RSTBRM command does not support optical files.
3. This command should not be used by control group *EXIT item processing because the results are unpredictable.

Notes:

1. You can restore data from an ADSM device by using this command. You can only specify one ADSM type device or *MEDCLS, which must select an ADSM device. The device selected can either be *APPC, which supports SNA network protocol, or *NET, which supports the TCP/IP protocol.
2. This command should not be used by control group *EXIT item processing as results will be unpredictable.

Example for RSTBRM

Example 1: Restoring All Integrated File System Objects

```
RSTBRM DEV(TAPE01)
```

This command allows you to select and restore integrated file system objects that were saved using the SAVBRM command and whose contents information still exist in the media contents information. In the example, the restore operation is for all integrated file system objects using TAPE01 to restore.

Error messages for RSTBRM

No error messages. <<

RSTOBJBRM (Restore Object Using BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.

RSTOBJBRM Command syntax diagram

Purpose

The Restore Object using BRM (RSTOBJBRM) command restores an object or several objects from the BRMS media content information. Any object that was saved by the Save Object using BRM (SAVOBJBRM) command or was saved as part of a control group can be restored by the RSTOBJBRM command. The types of objects that can be restored by this command are listed on the OBJTYPE parameter. They can be saved either as separate objects or as part of the library save operation. The RSTOBJBRM command restores the object description and contents of each object specified in the command.

Restrictions:

- If the user domain object user space (*USRSPC), user index (*USRIDX), or user queue (*USRQ) is restored to a library that is not permitted in the system value QALWUSRDMN (allow user objects in library), the object is converted to the system domain.
- >> You must have authority to the RSTOBJ command to use this command.
- You must have *USE authority to any auxiliary storage pool device specified for the RSTASP parameter. <<

Notes:

1. Do not precede an entry with an asterisk unless that entry is a “special value” that is shown (on the display itself or in the help information) with an asterisk.
2. Using the SAVLVL parameter allows you to restore selected copies of an object from the BRMS media content information. For instance, a recovery request for copy 1 of an object will restore the next to the most recent copy of the object on the BRMS media content information (current copy is the most recent copy).
3. You can restore data from a TSM server device by using this command. You can only specify one TSM device or *MEDCLS, which must select a TSM device. The TSM device selected can either be *APPC, which supports the SNA protocol, or *NET, which supports the TCP/IP protocol.

Note: This command should not be used by control group *EXIT item processing as results will be unpredictable.

Example for RSTOBJBRM

Example 1: Restoring an Object using BRMS

```
RSTOBJBRM OBJ(AP1000) SAVLIB(APLIB)
DEV(*MEDCLS) OBJTYPE(*FILE)
```

In the example the restore operation is for an object called AP1000, which is library APLIB. Any device that supports the media class assigned to the media containing AP1000 can be used in the restore operation. AP1000 is a file object type.

Error messages for RSTOBJBRM

None

RSTS36F (Restore System/36 File) Command Description

RSTS36F Command syntax diagram

Purpose

The Restore System/36 File (RSTS36F) command reads a file, creates a database physical or logical file (if it does not already exist), and copies any data into the file. This command restores to the system a single file or a group of files that are saved together using the System/36 save procedure. The input file can be a diskette file, tape file, or a database physical file on the iSeries 400. The file must have been created either on a System/36 using the SAVE system OCL procedure (or the equivalent OCL call of the \$COPY SSP utility), or on the iSeries 400 using the SAVS36F CL command.

The RSTS36F command accepts a diskette file created on a System/36, System/34, or System/32 using the \$COPY utility. The RSTS36F command accepts tape files created on a System/36. System/34 or System/32 diskette offline multiple-volume files are not accepted by this command.

The RSTS36F command accepts a diskette file created as a compressed file.

If the file being restored does not exist in the library specified on the TOLIB parameter, it is created. A physical file member is added using the name syntax 'Myymmdd', which identifies the original creation date of the file. This naming convention is needed by the System/36 environment in order to support date-differentiated files.

If a file name is specified for the TOFILE parameter, the name must meet the iSeries 400 naming standards. For more information about iSeries 400 naming conventions, see object naming rules.

If TOFILE(*SET) is specified, the files that are restored may have names that contain characters not allowed in an iSeries 400 simple object name. In this case, the file name is changed to an iSeries 400 extended name and the file is restored.

If the name contains a blank, a single quotation mark, a double quotation mark, an asterisk, a question mark, or a device control character (hexadecimal 00 through 3F or hexadecimal FF), the invalid characters are replaced with underlines. The file is then restored using the resulting simple or extended name; for example, A*_? would become "A___"). If a file already exists with this new name, it is handled like any other name (see the MBROPT parameter).

If a file name is changed because of invalid characters, an informational message (CPF 2C1F) is sent to the recursion level above the program that is running this command. If the name is changed from a simple name to an extended name, no message is sent.

If the restore function creates the file and the file was not previously secured, the new file is owned by the user issuing the RSTS36F command and the file is created with a default authority of *ALL (that is the same as AUT(*ALL)).

If the file was saved from the S/36 where the attributes were an extend value of zero or no value was specified, a default value of 32,767 divided by the record length is assigned. If an extend value of zero is required, use the Change Physical File (CHGPF) command (after the restore operation is completed) to set SIZE(*EXTEND) to zero. If the file was saved from the iSeries 400, the file is restored and the extend value does not change.

This function is intended only for exchanging files with a System/36.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The following authorities are required when running on a system using resource security:
 - *USE, *SECADM, and *ALLOBJ authorities for this command
 - *USE authority for the library specified in the TOLIB parameter
 - *CHANGE authority for the library specified in the TOLIB parameter when restoring a file that does not already exist on the system
 - *CHANGE and *OBJMGMT authority for the existing file (needed to add a new member) when restoring a date-differentiated physical file and a file by the same name but with a different creation date
 - *ALL authority for the file when restoring to an existing physical file with the same creation date and MBROPT(*REPLACE) specified
 - *CHANGE authority for the based-on physical file (this physical file was referred to as the *parent file* on System/36) if the file being restored is a System/36 alternative index file (that is, a logical file)
 - *USE authority for the diskette device description object and *USE authority for device file QSYSDKT in library QSYS, when restoring from diskette
 - *USE authority for the tape device description object and *USE authority for device file QSYSTAP in library QSYS, when restoring from tape
 - *USE authority for the file and *USE authority for the library that contains the file (PHYFILE parameter) if restoring from a database physical file
 - If the file doesn't exist on the system but a file authorization holder object by the same name does exist, *ALL authority or ownership for the authorization holder object
3. There is no replace function supported when restoring System/36 alternative index files (logical files). If restoring an alternative index file, no file object by the same name can already exist in the specified library.
4. If restoring a logical file, the based-on physical file must already exist in the library specified for the TOLIB parameter.
5. The iSeries 400 files that are the same as the System/36 date-differentiated files are multiple-member physical files. Date-differentiated alternative index files are not supported. The data for a physical file is stored in a member that is named using the syntax 'Myymmdd' where 'yymmdd' represents the original creation date (in year/month/day format) of the file.
Because all members of a physical file share the same file attributes (for example, record length and key information), date-differentiated files with the same name that are restored to the same library are required to have the same file attributes. If an attribute mismatch is present, the files are not restored.
6. Object-level and record-level functions, including read operations, should not be used for a file being restored by the RSTS36F command. If another operation is being done at the same time on the file (for example, moving the file or reading or adding records), the restore operation stops if it cannot allocate the file.

Required Parameters

TOFILE

Specifies the name of a single file that is restored to the system or that a group of files that are all saved at the same time using the System/36 SAVE procedure (a Save All Set) are to be restored to the system.

***SET:** A group of files from a Save All Set are restored to the system.

file-name: Specify the file name given to a single file when the file is restored to the system.

TOLIB Specifies the library that will contain the file being restored from the input file.

DEV Specifies the names of the devices used for the restore operation. Each device name must already be known on the system by a device description.

***PHYFILE:** The input file is the database physical file specified by the PHYFILE parameter.

device-name: Specify the name of the diskette device, the media library device, or the names of tape devices that are used for the operation. If more than one tape device is used, specify the names of the devices in the order in which they are used. When more than one tape volume is used, using more than one tape device permits one tape volume to be rewound or unloaded while another tape device processes the next tape volume. More information on device names is in the APPC, APPN, and HPR topic in the Information Center.

Optional Parameters

SET Specifies the name used to identify the Save All Set files saved on the diskette or tape by the SAVE procedure or the \$COPY utility on the System/36, System/34, or System/32.

#SAVE: The files are restored from a Save All Set with a set identifier of #SAVE.

set-identifier: Specify the set identifier of the Save All Set that is restored. Valid values range from 1 through 8 characters. The first character must be alphabetic (A through Z, #, \$, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except a comma (,), an apostrophe ('), or a blank.

IGCDTA

Specifies whether the file being restored can contain double-byte character set (DBCS) data.

***NO**: The file being restored cannot contain DBCS data. If a file already exists by the name specified on the TOFILE parameter in the specified library and the file allows DBCS data, an informational message is sent and the restore operation continues. The resulting file allows DBCS data (the file's IGC attributes are not changed by the restore operation).

***YES**: The file processes DBCS data.

FROMLABEL

Specifies the label of the diskette or tape file that contains the file being copied to the iSeries 400 (it identifies the input file for the restore operation). Up to 8 characters can be used to label the diskette or tape file.

For a group restore operation, this parameter can be used to specify the diskette or tape file label with a save all set where the restore operation is to begin. If no value is specified, the restore operation begins with the first file in the set.

If no value is specified, the value of the TOFILE parameter is used as the diskette or tape label. If the TOFILE parameter is an extended name, the characters between the quotation marks are used as the label. For example, specifying TOFILE("MIKE&BOB") and no FROMLABEL parameter would be the same as specifying FROMLABEL('MIKE&BOB').

CRTDATE

Specifies the creation date of the object.

The date must be entered in the job date format (DATFMT); if separators, specified by the job date separator character (DATSEP), are used, the value must be enclosed in apostrophes. The CRTDATE parameter cannot be specified if the FROMLABEL parameter is not specified.

For diskettes coming from a System/36, there may be multiple files on one diskette with the same name and different creation dates. Specifying a value for the CRTDATE parameter is the only way of choosing which diskette file to use.

For tapes from System/36 or iSeries 400, there can be multiple files on a tape with the same name. The user can use the CRTDATE parameter or SEQNBR parameter to choose the correct file. If a numeric SEQNBR parameter and the CRTDATE parameter are specified, the SEQNBR parameter takes precedence (if the file at the specified sequence number has a different creation date, the restore operation ends).

Specify the creation date of the tape or diskette file used for the restore operation. The date specified is changed to Julian format (cyyddd) for tape files and international format (yymmdd) for diskette files.

If no value is specified, the diskette file or tape file creation date is not used to select the input file. For diskette, take the first file in the diskette volume table of contents (VTOC) with the name specified on the FROMLABEL parameter. For tape, take the first file on the tape with the name specified on the FROMLABEL parameter. If the previous tape operation had specified ENDOPT(*LEAVE), the search for a matching file does not start at the beginning of the tape reel.

DATDIFF

Specifies whether the restore operation should allow date-differentiated files (that is, multiple files with the same name but different file creation dates).

***NO:** Date-differentiated files are not allowed.

If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with a different creation date (as determined by the member name), an error message is sent and the file is not restored.

If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with the same creation date (as determined by the member name), either the data in that member is replaced (if MBROPT(*REPLACE) is specified) or an error message is sent (if MBROPT(*NOREPLACE) is specified).

If no file exists by the name specified on the TOFILE parameter in the specified library, the file is restored normally.

***YES:** Date-differentiated files are allowed.

If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with a different creation date (as determined by the member name), a member is added to the file (only if the file attributes match), and the file's data is restored into the new member.

If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with the same creation date (as determined by the member name), either the data in that member is replaced (if MBROPT(*REPLACE) is specified) or an error message is sent (if MBROPT(*NOREPLACE) is specified).

If no file exists by the name specified on the TOFILE parameter in the specified library, the file is restored normally.

SEQNBR

Specifies, only when tape is used, which sequence number is used for the restore operation.

***SEARCH:** The volume in the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the data file is restored. If the last operation on the tape device specified ENDOPT(*LEAVE), the file search starts with the data file at the current tape position. Otherwise, the search starts with the first data file on the tape volume. The header label name from the file found by the SEQNBR parameter is compared to the name specified for the FROMLABEL parameter. If the names are the same, the restore operation can continue. If the names are different, the restore operation ends.

file-sequence-number: Specify the sequence number of the file that is used. Valid values range from 1 through 9999.

VOL

Specifies the volume identifiers of the tape volumes or diskette volumes from which the objects are being restored. The volumes must be in the same order as they were when the library was saved. If the library was known to the system before the restore operation, the system can verify whether the volumes put on tape contain the current version of the library. The volume that contains the beginning of the file to be restored should be placed in the tape or diskette unit. More information on this parameter is in commonly used parameters.

***MOUNTED:** The objects are restored from the volumes placed in the device specified on the DEV parameter.

volume-identifier: Specify the volume identifiers of the tapes or diskettes to be used for restoring the file. For each tape or diskette volume name, up to 6 characters can be specified using any combination of letters and digits. Up to 50 volume names can be specified.

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.

Note: This parameter is ignored if a diskette device is specified in the DEV parameter.

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

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

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

PHYFILE

Specifies the qualified name of the database physical file that is used as the input file for the restore operation.

The file must have a record length of 256 bytes.

If the specified file does not exist or is not a physical file, the file is not restored. If the physical file contains multiple members, the first member of the file is used.

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

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

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

library-name: Specify the name of the library to be searched.

file-name: Specify the name of the database file that is used as the input file for the restore operation.

MBROPT

Specifies whether the new records replace or are added to the existing records.

***NOREPLACE:** If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with the same creation date (as determined by the member name), an error message is sent and the data in that member is not replaced.

***REPLACE:** If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with the same creation date as the file being restored, the data in that member of the file is replaced.

Examples for RSTS36F

Example 1: Restoring From Diskette

```
RSTS36F TOFILE(ACCTRCV) TOLIB(QS36F) DEV(I1)
CRTDATE('01/22/85') VOL(SAVE1)
```

This command restores the file ACCTRCV into library QS36F. Assuming that I1 is the name of a diskette device description object, the file is restored from the diskette placed in the diskette device. The diskette must have a volume name of SAVE1. The diskette file used for the restore must have a file label of ACCTRCV and a creation date of January 22, 1985 (assuming the job date format is *MDY and the date separator is a '/').

Example 2: Restoring From Tape

```
RSTS36F TOFILE(PAY.VIEW) TOLIB(PAYLIB) DEV(T1)
        FROMLABEL('P*V') ENDOPT(*LEAVE)
```

The file P*V is restored from device T1 as a file named PAY.VIEW in library PAYLIB. Assuming T1 is a tape device, the file is copied from one or more tapes that are on device T1. No check is made on the tape volume name. When the restore operation ends, the tape is left positioned at the end of tape file P*V.

Example 3: Restoring from a Physical File

```
RSTS36F TOFILE(ACCTPAY) TOLIB(QS36F) DEV(*PHYFILE)
        PHYFILE(NETLIB/SENDFILE)
```

This command restores the file ACCTPAY in library QS36F from physical file SENDFILE in library NETLIB.

Example 4: Specifying Sequence Numbers

```
RSTS36F TOFILE(*SET) TOLIB(QS36F) DEV(T1 T2)
        SET(PAYFILES) FROMLABEL(FILE10) MBROPT(*REPLACE)
        DATDIFF(*YES) SEQNBR(*SEARCH) VOL(*MOUNTED)
        ENDOPT(*REWIND)
```

This command restores a subset of the files in the save all set called PAYFILES to library QS36F from tape. The restore operation begins with a tape file whose label is file 10. If one of the files being restored already exists in library QS36F with the same creation date as the saved file, the file is replaced. If a file already exists in library QS36F with a different creation date, a new date-differentiated number is added to the file. The restore operation uses the tape volumes that are placed in tape drives T1 and T2. After the restore operation is complete, the last tape volume is rewound to the beginning of the tape.

Error messages for RSTS36F

*ESCAPE Messages

CPF2C4A

Device &1 not correct for command.

CPF2C4B

Duplicate device &1 specified in device name list.

CPF2C4C

Diskette device &1 included in multiple device specification.

CPF2C4D

Not all files were restored.

CPF2C4E

Restore operation ended before all files were restored.

CPF2C45

Input file &1 cannot be processed by RSTS36F.

CPF2C47

Existing file &1 or member &3 in library &2 not replaced.

CPF2C48

Input file &1 in &2 not correct for command.

CPF2C49

Output file &1 in &2 not correct for command.

CPF2C5A

Alternate index file &1 in library &2 not replaced.

CPF2C5E

Input file &1 in &2 not correct for command.

CPF2C50

File description for file &1 is not available.

CPF2C52

Error occurred during attempt to create file &1 in library &2.

CPF2C53

Member &3 not added to file &1 in library &2 because error occurred.

CPF9810

Library &1 not found.

CPF9812

File &1 in library &2 not found.

CPF9814

Device &1 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

CPF9825

Not authorized to device &1.

CPF9826

Cannot allocate file &2.

CPF9830

Cannot assign library &1.

CPF9831

Cannot assign device &1.

CPF9845

Error occurred while opening file &1.

CPF9847

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

CPF9848

Cannot open file &1 in library &2 member &3.

CPF9849

Error while processing file &1 in library &2 member &3.

STATUS Messages*CPI2C11**

Copying records to file &1 in library &2 member &3.

RSTS36FLR (Restore System/36 Folder) Command Description

RSTS36FLR Command syntax diagram

Purpose

The Restore System/36 Folder (RSTS36FLR) command reads a diskette, tape, or database file that was prepared by the System/36 SAVEFLDR command. The data is converted so that it can be used by the iSeries 400. Both compressed and uncompressed data is supported when restoring data from a diskette.

Two types of System/36 folders are supported as input: document folders and interactive data definition utility (IDDU) folders.

- Document folders are converted to iSeries 400 folders, and the documents contained in them are converted to iSeries 400 documents. If a System/36 document is in the process of being edited when the SAVEFLDR is done, there are two existing versions of the document: an unchanged original, and one containing modifications. When this folder is restored on the iSeries 400, the unchanged original and the modification version are created. If the editing is completed by using OfficeVision on the iSeries 400, the modification version is removed.
- System/36 IDDU folders do not create folders on the iSeries 400. Instead, a library is created and the data dictionary is converted to an iSeries 400 data dictionary.

Notes:

1. When restoring a folder from the System/36, the user that issues the restore command becomes the owner and the one authorized to the folder.
2. When restoring a folder from the System/36 that contains an invalid name for the iSeries 400, the name is automatically changed to a different name generated by the system.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The user must have *USE authority to the from-folder, *CHG authority to the to-folder, and *USE authority to the device file or device description.
3. If the source folder is a document folder, the user must be enrolled in the system distribution directory and have either operational and data authorities to the folder or *ALLOBJ special authority. If replacing a document, the user must have *ALL authority to the document.

Required Parameters

FROMFLR

Specifies the name of the folder that is being restored. Specify the name of the folder.

DEV

Specifies the name of the device from which to restore the specified folder. The device name must be known to the system by a device description.

***PHYFILE:** The folder is in a physical database file. If DEV(*PHYFILE) is specified, then the PHYFILE parameter must be specified.

device-name: Specify the name of the diskette device, the media library device, or the names of tape devices that are used for the operation. If more than one tape device is used, specify the names of the devices in the order in which they are used. When more than one tape volume is used, using more than one tape device permits one tape volume to be rewound or unloaded while another tape device processes the next tape volume. More information on device names is in the APPC, APPN, and HPR topic in the Information Center.

Optional Parameters

TOFLR

Specifies the name of the folder in which to put the restored folder's contents. This parameter is used only for document folders. An error message is sent if this parameter is specified for any other type of folder.

***FROMFLR:** The old name of the folder specified in the FROMFLR parameter is used for the name of the restored folder.

folder-name: Specify the new name for the folder being restored.

TODTADCT

Specifies the name of the library where an IDDU dictionary (folder) is restored. This parameter is used only for IDDU dictionaries. An error message is sent if this parameter is specified for any other type of folder. When this parameter is specified, a library is created and a data dictionary is created in the new library. An error message is sent if a library with the specified name already exists.

***FROMFLR:** The old name of the folder specified in the FROMFLR parameter is used for the name of the restored library.

library-name: Specify the name of the library where the restored folder is placed.

PHYFILE

Specifies the qualified name of the database file that contains the System/36 folder.

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

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

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

library-name: Specify the name of the library to be searched.

file-name: Specify the name of the database file that contains the System/36 folder.

MBR Specifies the name of the file member being restored.

***FIRST:** The first member in the database file is used.

member-name: Specify the name of the file member that contains the restored System/36 folder.

CRTDATE

Specifies the creation date of the object.

The date must be entered in the job date format (DATFMT); if separators, specified by the job date separator character (DATSEP), are used, the value is enclosed in apostrophes.

Diskettes from a System/36 may contain several files with identical names and different creation dates. Specifying a value for the CRTDATE parameter is the only way to choose which diskette file to use.

Tapes from a System/36 or iSeries 400 may contain several files with identical names and different creation dates. The user can use the CRTDATE or SEQNBR parameter to choose the correct file. If both a numeric SEQNBR parameter and the CRTDATE parameter are specified, the SEQNBR parameter takes precedence. If the file at the specified sequence number has a different creation date, the restore operation ends.

Specify the creation date of the tape or diskette file used for the restore operations. The date specified is changed to Julian format (cyyddd) for tape files and international format (yymmdd) for diskette files.

VOL Specifies the volume identifiers of the tape volumes or diskette volumes from which the objects are being restored. The volumes must be in the same order as they were when the library was saved. If the library was known to the system before the restore operation, the system can verify whether the volumes put on tape contain the current version of the library. The volume that contains the beginning of the file to be restored should be placed in the tape or diskette unit. More information on this parameter is in commonly used parameters.

***MOUNTED:** The objects are restored from the volumes placed in the device specified on the DEV parameter.

volume-identifier: Specify the identifier of one or more volumes in the order in which they are used. Up to 6 characters identify each volume.

SEQNBR

Specifies the sequence number of the file being restored.

***SEARCH:** The tape is searched for the first sequence number that contains the specified folder.

sequence-number: Specify the sequence number of the file. Valid values range from 1 through 9999.

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.

Note: This parameter is ignored if a diskette device is specified in the DEV parameter.

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

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

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

KBDTYPE

Specifies the character set and code page of the System/36 that created the data in the folder.

***SYSVAL:** The System/36, from which the folder was saved, uses the same character set and code page as the iSeries 400.

***MULTI:** The System/36, from which the folder was saved, is configured as a multinational system.

character-set-identifier: Specify one of the identifiers from the Keyboard Mapping table if the System/36 is a non-multinational system and the iSeries 400 is not using the same character set and code page as the System/36. On the System/36, a list of language or country or region names is presented from which to make a selection. If multinational is chosen, the System/36 maps it into a single code regardless of language. For example, if Denmark is chosen and not multinational, use DMB with RSTS36FLR. If Denmark is chosen with multinational on the System/36, use *MULTI with RSTS36FLR.

Table 1. Keyboard Mapping

Language/Country or Region	Identifier	ASCII Device Group
Albania	ALI	
Arabic X/Basic	CLB	D

Language/Country or Region	Identifier	ASCII Device Group
Austria/Germany	AGB	A, B
Austria/Germany Euro Currency	AGE	
Austria/Germany Multinational	AGI	A, B
Belgium Multinational	BLI	B
Belgium Multinational Euro Currency	BLM	
Brazilian Portuguese	BRB	
Brazilian Portuguese Euro Currency	BRE	
Bulgaria	BGB	
Canadian French	CAB	A, B
Canadian French Multinational	CAI	A, B
Canadian French Multinational Euro Currency	CAM	
Chinese (Simplified)	RCB	
Chinese (Traditional)	TAB	
Croatia	YGI	
Cyrillic	CYB	
Czech Republic	CSB	
Denmark	DMB	B
Denmark Euro Currency	DME	
Denmark Multinational	DMI	B
Estonia	ESB	
Finland/Sweden	FNB	B
Finland/Sweden Euro Currency	FNE	
Finland/Sweden Multinational	FNI	B
France (Azerty)	FAB	A, B
France (Azerty) Euro Currency	FAE	
France (Azerty) Multinational	FAI	A, B
France (Qwerty)	FQB	
France (Qwerty) Multinational	FQI	
Greece ¹	GNB ¹	
Hebrew	NCB	D
Hungary	HNB	
Iceland	ICB	
Iceland Euro Currency	ICE	
Iceland Multinational	ICI	
International	INB	
International Multinational	INI	
Iran (Farsi)	IRB	
Italy	ITB	A, B
Italy Euro Currency	ITE	
Italy Multinational	ITI	A, B
Japan English	JEB	
Japan English Multinational	JEI	
Japan Kanji (for PS/55 and 5295 display stations)	JKB	
Japan Latin Extended	JPB	
Japan United States Basic	JUB	
Japan Katakana (for 5251, 5291, 5292, and 3180 Katakana display stations)	KAB	
Korea	KOB	
Latin-2/ROECE	ROB	
Latvia	LVB	
Lithuania	LTB	
FYR Macedonia (Former Yugoslav Republic)	MKB	
Netherlands	NEB	
Netherlands Euro Currency	NEE	

Language/Country or Region	Identifier	ASCII Device Group
Netherlands Multinational	NEI	
Norway	NWB	B
Norway Euro Currency	NWE	
Norway Multinational	NWI	B
Poland	PLB	
Portugal	PRB	B
Portugal Euro Currency	PRE	
Portugal Multinational	PRI	B
Romania	RMB	
Russia	RUB	
Serbia (Cyrillic)	SQB	
Serbia (Latin)	YGI	
Slovakia	SKB	
Slovenia	YGI	
Spain	SPB	B
Spain Euro Currency	SPE	
Spain Multinational	SPI	B
Spanish Speaking	SSB	B
Spanish Speaking Multinational Euro Currency	SSE	
Spanish Speaking Multinational	SSI	B
Sweden	SWB	B
Sweden Euro Currency	SWE	
Sweden Multinational	SWI	B
Switzerland/French Multinational	SFI	B
Switzerland/French Multinational Euro Currency	SFM	
Switzerland/German Multinational	SGI	B
Switzerland/German Multinational Euro Currency	SGM	
Thai	THB	
Turkey (Qwerty)	TKB	
Turkey (F)	TRB	
Ukraine	UAB	
United Kingdom	UKB	A, B
United Kingdom Euro Currency	UKE	
United Kingdom Multinational	UKI	A, B
United States/Canada	USB	A, B, C
United States/Canada Euro Currency	USE	
United States/Canada Multinational	USI	A, B, C
Urdu	PKB	
Vietnam	VNB	
Languages of the former Yugoslavia	YGI	

Note:

- ¹ The GNB code is the current identifier for Greece. The GKB code was used prior to V2R1, and continues to be supported, but provides fewer characters than the recommended GNB code.

DUPDOC

Specifies whether existing documents are renamed or replaced. This parameter is ignored when restoring a data dictionary.

***RENAME:** If a folder already exists with the name specified in the TOFLR parameter and a document in that folder exists with the same name as an incoming document, the incoming document is created in the folder with a new system-generated name. All other documents are created in the specified folder.

***REPLACE:** If a folder already exists by the name specified in the TOFLR parameter, and a document in that folder exists with the same name as an incoming document, the existing document is replaced. All other documents are created in the specified folder.

Examples for RSTS36FLR

Example 1: Restoring From Diskette

```
RSTS36FLR FROMFLR(TXTDRA) DEV(DKT01)
  CRTDATE('01/16/88') VOL(SAVE1)
```

This command restores the folder TXTDRA. Assuming that DKT01 is the name of a diskette device description, the folder is restored from the diskette placed in the diskette device. The diskette must have a volume name of SAVE1 and the folder must have a creation date of January 16, 1988.

Example 2: Restoring From Tape

```
RSTS36FLR FROMFLR(TXTMEL) DEV(TAP1)
  TOFLR(TXTNEW) SEQNBR(7)
  ENDOPT(*UNLOAD)
```

This command restores the folder TXTMEL to a folder named TXTNEW. Assuming TAP1 is a tape device, the folder is restored from one or more tapes that are on device TAP1. The folder must have a sequence number of 7. The tape is unloaded from the device at the end of processing.

Example 3: Restoring to a Data Dictionary

```
RSTS36FLR FROMFLR(DONDCT) DEV(*PHYFILE)
  TODTADCT(NEWDC) PHYFILE(MYLIB/MYFILE)
  MBR(THISONE)
```

This command restores the folder DONDCT to a data dictionary named NEWDC. Assuming that MYFILE is a physical file located in library MYLIB and contains the member THISONE, the folder is restored from member THISONE.

Error messages for RSTS36FLR

*ESCAPE Messages

CPF8D09

Folder &1 not restored.

RSTS36LIBM (Restore System/36 Library Members) Command Description

RSTS36LIBM Command syntax diagram

Purpose

The Restore System/36 Library Members (RSTS36LIBM) command reads a file containing library members, creates database source or data file members on the iSeries 400, and copies the member data from the file into each restored member. The input file can be a diskette file, tape file, or database physical file on the iSeries 400. The file could have been created on a System/36, a System/34, or a System/32 using either the FROMLIBR or SAVELIBR system OCL procedure (or the equivalent call of \$MAINT), or on an iSeries 400 using the SAVS36LIBM CL command.

Diskette files created on a System/34 using the BACKUP procedure or \$BACK utility are not accepted by this command. Compressed SAVELIBR diskette files (used by IBM to distribute system libraries for System/36 after release 5.0) are not accepted by this command.

In System/36 terms, the input file format could be a SAVELIBR diskette file or tape file, a record-mode LIBRFILE diskette file, tape file, or physical file, or a sector-mode LIBRFILE diskette file, tape file, or physical file. In other words, the input file can be any diskette file, tape file, or physical file created by the System/36 \$MAINT SSP utility.

If the library identified by the TOLIB parameter value does not exist, it is created. Also, the source files QS36SRC and QS36PRC are created if they do not exist in the restore-to library.

If the restore operation creates the library, the new library is owned by the user running the RSTS36LIBM command and the library is created with a default authority of *ALL (that is, the same as AUT(*ALL)).

If a sector-mode FROMLIBR file or a SAVELIBR file created on a System/36 is being restored, data files QS36LOD and QS36SBR may also be created to hold restored load and subroutine members. Restored load and subroutine members are not converted on the current system.

Restrictions:

- The following authorities are required when running on a system using resource security:
 1. *SECADM and *ALLOBJ authorities
 2. *USE authority for this command and *USE authority for the Create Library (CRTLIB) command if the library needs to be created
 3. *USE authority for the CRTSRCPF command if QS36SRC or QS36PRC must be created
 4. *USE authority for the CRTPF command if QS36LOD or QS36SBR must be created
 5. *CHANGE authority for the library specified in the TOLIB parameter
 6. *CHANGE and *OBJMGMT authority for file QS36SRC in the specified library if restoring source library members
 7. *CHANGE and *OBJMGMT authority for file QS36PRC in the specified library if restoring procedure library members
 8. *CHANGE and *OBJMGMT authority for file QS36LOD in the specified library if restoring load library members
 9. *CHANGE and *OBJMGMT authority for file QS36SBR in the specified library if restoring subroutine library members
 10. *USE authority for the diskette device description object, *USE authority for device file QSYSDKT in library QSYS if restoring from diskette
 11. *USE authority for the tape device description object and *USE authority for device file QSYSTAP in library QSYS, if restoring from tape
 12. *USE authority for the file and *USE authority for the library that contains the file (PHYFILE parameter) if restoring from a database physical file
- No object-level or record-level operations should be active for files QS36SRC, QS36PRC, QS36SBR, and QS36LOD while members are being restored by RSTS36LIBM. If the necessary files cannot be allocated exclusively, no members are restored.
- The member name or generic member name specified (TOMBR parameter) must meet iSeries 400 naming standards.

If a generic member name or *ALL is specified, a member may be selected to be restored that has a name containing characters not allowed in an iSeries 400 *simple* object name. In this case, the member name is restored using the iSeries 400 *extended* name syntax (for example, A!B would become 'A!B').

If the name contains a blank, a single quotation mark (') a double quotation mark ("), an asterisk (*), a question mark (?), or a device control character (hexadecimal '00'-'3F' or hexadecimal 'FF'), these characters are replaced by underlines and the member is restored using the resulting simple or extended name (for example, A*/? would become A_/_ and A? would become A_).

An informational message is sent each time invalid characters are replaced to get a valid name. An additional informational message is sent if the resulting name change caused a member to be replaced. No message is sent if a member is restored using the extended name syntax without replacing invalid characters.

Required Parameters

TOMBR

Specifies the names of the members to restore.

***ALL:** All members of the specified member types are restored.

member-name: The members that have the specified member name are restored.

generic-member-name:* All members that have the specified generic member name are restored. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. For more information on the use of generic names, refer to generic names.

TOLIB Specifies the library that will contain the members to restore from the input file.

DEV Specifies the names of the devices to use for the restore operation. Each device name must already be known on the system by a device description.

***PHYFILE:** The input file is the database physical file specified by the PHYFILE parameter.

device-name: Specify the name of the diskette device, the media library device, or the names of tape devices that are used for the operation. If more than one tape device is used, specify the names of the devices in the order in which they are used. When more than one tape volume is used, using more than one tape device permits one tape volume to be rewound or unloaded while another tape device processes the next tape volume. More information on device names is in the APPC, APPN, and HPR topic in the Information Center.

Optional Parameters

SRCMBRS

Specifies which source member types (source and procedure members on System/36) are restored.

***ALL:** All source and procedure members in the input file that match the member name specified (TOMBR parameter) are restored. For example, if TOMBR(PAY*) and SRCMBRS(*ALL) is specified, all source and procedure members that start with the letters 'PAY' are restored.

***SRC:** Only System/36 source members (to file QS36SRC) that match the member name specified (TOMBR parameter) are restored.

***PRC:** Only System/36 OCL procedure members (to file QS36PRC) that match the member name specified (TOMBR parameter) are restored.

***NONE:** No source or procedure library members are restored.

OBJMBRS

Specifies which object member types (load and subroutine members on System/36) are restored. Because the System/36 and the iSeries 400 are not object compatible, any restored members are not immediately useable after the restore operation. IBM-supplied commands or user-written operations must be run to convert the object member to a useable iSeries 400 object.

***NONE:** No System/36 object members are restored.

***SBR:** Only System/36 subroutine members (to file QS36SBR), which match the member name specified (TOMBR parameter), are restored.

***LOD:** Only System/36 load members (to file QS36LOD), which match the member name specified (TOMBR parameter), are restored.

***ALL:** All load and subroutine members in the input file which match the member name specified (TOMBR parameter), are restored. For example, if TOMBR(CONF1) and OBJMBRS(*ALL) are specified, load and subroutine members with the member name 'CONF1' are restored.

MBROPT

Specifies, for database files currently on the system, which file members are restored.

***NEW:** Only new members (members that do not already exist in the appropriate source or data file) are restored.

***OLD:** Only old members (members that already exist in the appropriate file) are restored. The existing member is replaced by the copy of the member restored from the file.

***ALL:** All members are restored, even if they already exist in an iSeries 400 source or data file. Members that do not already exist are created, and members that do exist are replaced.

IGCDTA

Specifies whether the source and procedure members being restored can contain double-byte character set (DBCS) data.

Note:

This attribute is used if the restore operation needs to create source files QS36SRC and QS36PRC to hold the restored library members. If the QS36SRC or QS36PRC source file already exists in the library specified on the TOLIB parameter and the file's DBCS capability does not match the parameter, an error message is sent and no member is restored.

***NO:** The file does not process DBCS data.

***YES:** The file processes DBCS data.

FROMLABEL

Specifies the label (8 characters maximum) of the diskette file or tape file that contains the members to copy to the iSeries 400 (it identifies the input file for the restore operation). If the DEV parameter value is not *PHYFILE, a FROMLABEL value must be specified.

CRTDATE

Specifies the creation date of the object.

The date must be entered in the job date format (DATFMT); if separators are used, specified by the job date separator character (DATSEP), the value must be enclosed in apostrophes.

For diskettes coming from a System/36, there may be several files on one diskette with the same name and different creation dates. Specifying a value for the CRTDATE parameter is the only way of choosing which diskette file to use.

For tapes from System/36 or the iSeries 400, there can be several files on a tape with the same name. The user can use the CRTDATE or SEQNBR parameter to choose the correct file. If a numeric SEQNBR parameter and the CRTDATE parameter are specified, the SEQNBR parameter takes precedence. If the file at the specified sequence number has a different creation date, the restore operation ends.

Specify the creation date of the tape file or diskette file to use for the restore operation. The date specified is changed to Julian format (cyyddd) for tape files and international format (yymmdd) for diskette files.

If no value is specified, the diskette file or tape file creation date is not used to select the input file. For diskette files, take the first file in the diskette VTOC with the name specified on the

FROMLABEL parameter. For tape files, take the first file on the tape with the name specified on the FROMLABEL parameter. If the previous tape operation had specified ENDOPT(*LEAVE), the search for a matching file does not start at the beginning of the tape reel.

SEQNBR

Specifies, only when tape is used, which sequence number is used for the restore operation.

***SEARCH:** The volume that is placed in the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the data file is restored. If the last operation on the tape device specified ENDOPT(*LEAVE), the file search begins with the data file at the current tape position. Otherwise, the search begins with the first data file on the tape volume. The header label name from the file found by the SEQNBR parameter is compared to the name specified for the FROMLABEL parameter. If the names are the same, the restore operation can continue. If the names are different, the restore operation ends.

file-sequence-number: Specify the sequence number of the file that is used. Valid values range from 1 through 9999.

VOL

Specifies the volume identifiers of the tape volumes or diskette volumes from which the objects are being restored. The volumes must be in the same order as they were when the library was saved. If the library was known to the system before the restore operation, the system can verify whether the volumes put on tape contain the current version of the library. The volume that contains the beginning of the file to be restored should be placed in the tape or diskette unit. More information on this parameter is in commonly used parameters.

***MOUNTED:** The objects are restored from the volumes placed in the device specified on the DEV parameter.

volume-identifier: Specify the volume identifiers of the tapes or diskettes to be used for restoring the file. For each tape or diskette volume name, up to 6 characters can be specified using any combination of letters and digits. Up to 50 volume identifiers can be specified.

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.

Note:

This parameter is ignored if a diskette device is specified in the DEV parameter.

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

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

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

PHYFILE

Specifies the qualified name of the database physical file that is used as the input file for the restore operation.

The file must have a record length of between 40 and 120 bytes if it contains members saved in record mode. The record length must be 256 if it contains members saved on a System/36 in sector mode.

If the specified file does not exist or is not a physical file, no library members are restored. If the physical file contains several members, the first member of the file is used.

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

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

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

library-name: Specify the name of the library to be searched.

file-name: Specify the name of the database physical file that is used as the input file for the restore operation.

Examples for RSTS36LIBM

Example 1: Restoring All Members

```
RSTS36LIBM TOMBR(XYZ1) TOLIB(JOHNSON) DEV(I1)
SRCMBRS(*PRC) MBROPT(*ALL) FROMLABEL('XYZ1')
```

The single OCL procedure member XYZ1 is restored as a member of source file QS36PRC in library JOHNSON. Assuming I1 refers to a diskette device, the input diskette file must have the label XYZ1.

Example 2: Specifying a Generic Member Name

```
RSTS36LIBM TOMBR(X*) TOLIB(ORDER) DEV(*PHYFILE)
PHYFILE(NETLIB/S36SRC)
```

All source and procedure members with names starting with the character 'X' and that do not already exist as members of QS36SRC and QS36PRC in library ORDER are restored. The members are restored from file S36SRC in library NETLIB.

Error messages for RSTS36LIBM

*ESCAPE Messages

CPF2C4A

Device &1 not correct for command.

CPF2C4B

Duplicate device &1 specified in device name list.

CPF2C4C

Diskette device &1 included in multiple device specification.

CPF2C41

&2 members restored, &3 members not restored, &4 members excluded.

CPF2C42

No members restored to library &1, &4 members excluded.

CPF2C46

Input file &1 cannot be processed by RSTS36LIBM.

CPF2C48

Input file &1 in &2 not correct for command.

CPF2C49

Output file &1 in &2 not correct for command.

CPF2C5E

Input file &1 in &2 not correct for command.

CPF2C50

File description for file &1 is not available.

CPF2C52

Error occurred during attempt to create file &1 in library &2.

CPF2C53

Member &3 not added to file &1 in library &2 because error occurred.

CPF2C57

DBCS attribute not same as existing file &1 in &2.

CPF2C70

Input file &1 contains incorrect control statement record.

CPF2C71

NAME keyword on COPY statement missing or incorrect.

CPF2C72

LIBRARY keyword on COPY statement missing or incorrect.

CPF9807

One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

CPF9810

Library &1 not found.

CPF9812

File &1 in library &2 not found.

CPF9814

Device &1 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

CPF9824

Not authorized to command &1 in library &2.

CPF9825

Not authorized to device &1.

CPF9826

Cannot allocate file &2.

CPF9830

Cannot assign library &1.

CPF9845

Error occurred while opening file &1.

CPF9847

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

CPF9849

Error while processing file &1 in library &2 member &3.

***STATUS Messages**

RSTUSFCNR (Restore Ultimedia System Facilities Container)

Command Description

RSTUSFCNR Command syntax diagram

Purpose

The Restore Ultimedia System Facilities Container (RSTUSFCNR) command restores to the system the multimedia objects that have been saved using the Save Ultimedia System Facilities Container (SAVUSFCNR) command. The multimedia objects that can be restored using this command are the following:

- Multimedia objects that have been copied to a shared folder container
- Document library objects (DLOs) that hold the Ultimedia Facilities data stream files.

Restriction: Only those objects that have been placed in an Ultimedia Facilities container and whose data stream files are in the Ultimedia Facilities repository when they were saved are restored. If the data stream file was not in the Ultimedia Facilities repository at the time of the save, only the object's DLO is restored.

Required Parameters

SAVCNR

Specifies the identifier of the multimedia container object that is to be restored. This is the identifier of the container object saved with the SAVUSFCNR command.

RSTCNR

Specifies the identifier of the multimedia container object to which the saved container is to be restored.

***SAVCNR:** The container objects are restored to the container with the same identifier as the container in which they were saved. If a container with this identifier does not exist on the target system, the container is created as a first level container under the QUMBREPO folder.

container-identifier: Specify the identifier of the container to which the saved container objects are to be restored from tape. If the container specified does not exist, an error message is sent to the job log and processing ends for this command.

DEV Specifies the name of the device from which the multimedia objects are to be restored.

***SAVF:** The objects are restored from the save file specified on the SAVF parameter.

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

tape-device-name: Specify the names of one or more tape devices used for the restore operation. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume. You can specify a maximum of four tape devices.

Optional Parameters

VOL Specifies the volume identifier of the tape from which the multimedia objects are to be restored.

***MOUNTED:** The objects are restored from the volumes placed in the device specified on the DEV parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are put on the device and used. Each volume identifier contains a maximum of 6 alphanumeric characters. A blank is used as a separator character when listing multiple identifiers. You can specify a maximum of 75 volumes.

SEQNBR

Specifies the starting sequence number of the files on the tape used for the restore operation.

***SEARCH:** The volume on the device is searched for a data file containing the multimedia objects to be restored; when a match is found, the data file is searched for the multimedia objects to be restored. If the specified multimedia objects are not found in that data file, the search continues for the next data file that matches the value specified on the LABEL parameter.

Note:

If the last operation on the device specified the same volume and ENDOPT(*LEAVE) (that is, the specified tape is positioned at the location at which the last operation ended), the search operation starts with the first data file beyond the current tape position. The files located in front of the current tape position are not searched.

sequence-number: Specify the sequence number of the file to be used as the starting point for the restore operation.

LABEL

Specifies the label of the container object that is used to find the file that was written to tape during the save operation.

***SAVCNR:** The container label is the same as the last 17 characters of the Ultimedia Facilities container identifier specified on the SAVCNR parameter.

data-file-identifier: Specify the data file identifier which is the label of the data file that contains the multimedia data to be restored.

ENDOPT

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

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

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

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

SAVF Specifies the qualified name of file that contains the multimedia objects that are being restored.

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

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

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

library-name: Specify the name of the library to be searched.

save-file-name: Specify the name of the save file to restore.

SAVDATE

Specifies the date on which the container was saved. If more than one version of the container exists on the media, this parameter identifies which version of the container to restore. If this parameter is not specified, the version of the multimedia objects being restored is the first version found on the volume or, if the LABEL parameter is specified, the version found with the specified label.

The date must be specified in the system date format specified by the system value QDATFMT; if the separators specified by the system value QDATSEP are used, the value must be enclosed in apostrophes.

SAVTIME

Specifies the time when the container was saved. If more than one version of the container exists on the media, this parameter identifies which version of the container is restored. If this parameter is not specified, the version of the multimedia objects being restored is the first version found on the volume or, if the LABEL parameter is specified, the version found with the specified file label.

The time must be specified as a 6-digit value, in the format hours, minutes, and seconds (hhmmss). If colons are used as separators, the value must be enclosed in apostrophes.

Note: This parameter is valid only if SAVDATE is specified.

ALWOBJDIF

Specifies whether the following differences that may be encountered during a restore operation are allowed.

- The byte stream for a multimedia object resides within the Ultimedia Facilities repository on the target system but is not on the tape.
- The byte stream for a multimedia object is on the tape but is not stored within the Ultimedia Facilities object repository on the target system.
- The members of a multimedia key object on the tape are different than those on the target system.
- The members of a multimedia sequence object on the tape are different than those on the target system.

***ALL:** The differences listed are allowed for the restore operation. An informational message is sent and the multimedia object is restored.

***NONE:** The differences listed are not allowed on the restore operation. A diagnostic message is sent for the multimedia object, and the multimedia object is not restored.

Example for RSTUSFCNR

```
RSTUSFCNR SAVCNR(SEMINARVIDEOSPOT0004)
RSTCNR(*SAVCNR) DEV(TAP01)
```

This command restores the multimedia container object with the container identifier SEMINARVIDEOSPOT0004 to the multimedia container object with the same container identifier. The container is restored from the tape unit TAP01.

No error messages.

RSTUSRPRF (Restore User Profiles) Command Description

RSTUSRPRF Command syntax diagram

Purpose

The Restore User Profiles (RSTUSRPRF) command restores the basic parts of a user profile or a set of user profiles saved by the Save System (SAVSYS) or the Save Security Data (SAVSECDTA) commands. The RSTUSRPRF command restores only the special authority given in the Create User Profile (CRTUSRPRF) command; it does not restore the authority for the named objects owned by other users. To do that, the Restore Authority (RSTAUT) command must be used after the profiles, libraries, and objects are restored. If all user profiles are restored, the authorization lists and authority holders that existed when the SAVSYS command or SAVSECDTA command was run are also restored.

Before this command is specified, all other operations on the system must be stopped if USRPRF(*ALL) is specified. This requires ending all subsystems through the End Subsystem (ENDSBS(*ALL)) command or End System (ENDSYS) command or specifying this command when the operating system is started. The RSTUSRPRF command is normally used after the restore of the operating system but before the user libraries are restored. The user profiles must be restored before any libraries or objects belonging to them can be restored. After the libraries and their objects are restored, the authority for the objects is restored to the user profiles by the RSTAUT command. At the completion of the command, either message CPF3775 or message CPC3705 is sent to QHST. More information on restoring the system is in the Backup, Recovery, and Availability topic in the Information Center.

The following situations may apply to user profiles being restored by the RSTUSRPRF command:

- If a user profile exists on the system, but not on the media, the system profile remains.
- If a user profile exists on the media, but not on the system, a new user profile is created.
- If the user profile exists on both the media and the system, the media user profile is restored.
- If user profiles exist on the system, there are no changes to the existing object authorities.

The following applies to restoring passwords and group connections for restored user profiles:

- The default action when user profiles are restored individually is as follows:
 - If the user profile exists on the media but not on the system, then the new user profile is created without its passwords or group connection.
 - If the user profile exists on both the media and the system, the media user profile is restored. However, the password and group connection from the system remains unchanged.
- If all user profiles are restored or if user profiles are restored individually and SECDDTA(*PWDGRP) is specified, then the group connections are also restored from the media.
- If all user profiles are restored or if user profiles are restored individually and SECDDTA(*PWDGRP) is specified, then restoring the password depends upon the value of the QPWDLVL system value. The possible values of QPWDLVL are as follows:
 - QPWDLVL value 0, all passwords are restored from the media.
 - QPWDLVL value 1, all passwords except the password used by the Netserver product are restored from the media.
 - QPWDLVL value 2, all passwords are restored from the media.
 - QPWDLVL value 3, only new encrypted passwords generated at V5R1M0 or later are restored from the media. Any passwords from an earlier release will not be restored.

Note: This command ignores all file overrides that are currently in effect for the job.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The user of this command must have *SAVSYS authority in the user profile, specified by SPCAUT(*SAVSYS).

3. Before this command is specified, all other operations on the system must be ended if USRPRF(*ALL) is specified. The End System (ENDSYS) or End Subsystem (ENDSBS) command can be used to end these operations. You must have *JOBCTL authority to use the ENDSYS or ENDSBS command.
4. To restore authorization lists and authority holders, USRPRF(*ALL) must be specified.
5. When SECDDTA(*PVTAUT) is specified, the user profile must exist on the system. Internal private authority objects will not be created for a non-existing user profile.
6. The user must have both *ALLOBJ and *SECADM special authorities in order to specify SECDDTA(*PWDGRP).

Required Parameter

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

***SAVF:** The save file specified on the SAVF parameter is used for the restore operation.

diskette-device-name: Specify the name of the diskette device used to restore the user profiles.

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

tape-media-library-device-name: Specify the name of the tape media library device used to restore the user profiles.

tape-device-name: Specify the names of one or more tape devices used for the restore operation. If more than one tape device is used, specify the names of the devices in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Optional Parameters

USRPRF

Specifies the names of one or more user profiles to restore. To be restored, the user profiles must exist on the media from the Save System (SAVSYS) or Save Security Data (SAVSECDDTA) commands.

***ALL:** All of the user profiles, authorization lists, authority holders, and digital certificate manager information saved by the SAVSYS or SAVSECDDTA commands are restored.

***NEW:** All of the user profiles, authorization lists, and authority holders saved by the SAVSYS or SAVSECDDTA commands which do not currently exist on the system are restored.

» ***NONE:** No user profiles are restored. «

generic-user-profile-name:* Specify one or more generic names of sets of user profiles to restore. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. For more information on the use of generic functions, refer to generic functions.

user-profile-name: Specify one or more names of specific user profiles that are restored. Both generic names and specific names can be specified in the same command.

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

Note:

For tape, the user profiles are restored from the volumes placed on the tape devices, in the order that those devices were specified. This operation is normally performed immediately after a restore of the system is performed and the user profiles are on the same volume.

***MOUNTED:** The objects are restored from the volumes placed in the device specified on the DEV parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

volume-identifier: Specify the identifiers of up to 75 volumes in the order in which they are placed in the devices and used to restore the user profiles.

SEQNBR

Specifies the sequence number of the tape file used for the restore process.

***SEARCH:** The volume placed in the device is searched for a file containing the saved user profiles; when a match is found, the user profiles are restored. If a match is not found, you must load another tape and try the command again. If the last operation on the device specified ENDOPT(*LEAVE) (the tape is positioned at the location where the last operation ended), the file search starts with the first file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last operation (or if the tape was manually rewound since an ENDOPT(*LEAVE) operation), the search begins with the first file on the volume.

sequence-number: Specify the sequence number of the file. Valid values range from 1 through 16777215.

ENDOPT

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

Note:

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

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

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

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

SAVF Specifies the save file that contains the data from which to restore objects.

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

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

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

library-name: Specify the name of the library to be searched.

save-file-name: Specify the name of the file used to restore user profiles.

MAIL Specifies whether the OfficeVision distribution objects saved from a release before V2R2M0 are restored.

Note: You can specify *YES on this parameter only if you specify *ALL on the USRPRF parameter.

***NO:** Distribution objects that are part of your mail are not restored with the user profile.

***YES:** Distribution objects that are part of your mail are restored with the user profile if the save data was created before release V2R2M0. Otherwise, no distribution objects are restored. For saved distribution objects created on V2R2M0 or later, specify DLO(*MAIL) on the Restore Document Library Objects (RSTDLO) command to restore your mail.

ALWOBJDIF

Specifies whether certain differences encountered during a restore are allowed. There are two differences for this command:

- The owner of the object on the system is different than the owner of the object from the save.
- The primary group of the object on the system is different than the primary group of the object from the save.
- The object is secured by an authorization list and is being restored to a system other than the one on which it was saved.

Note: In order to use this parameter, *ALLOBJ authority is required.

***NONE:** The differences described above are not allowed for the restore operation. For an ownership difference, the object is not restored. For primary group difference, the object is not restored. For an authorization list difference, the object is restored, but the object is not linked to the authorization list, and public authority is set to *EXCLUDE.

***ALL:** The differences previously described are allowed for the restore operation. The object is restored. The following should be noted:

- If the owners of the object do not match, the object will be restored, but the ownership and authorities it had before the restore operation are retained.
- If the primary groups of the object do not match, the object will be restored, but the primary group it had before the restore operation is retained.
- The informational message causes a diagnostic message to be sent indicating that security or integrity changes occurred during the restore. This results in an escape message for the restore function.
- If the user is restoring objects to a system different from the one on which they were saved and the objects are secured by an authorization list, specifying *ALL automatically links the objects to the authorization list again. If the authorization list does not exist on the new system, a message that includes the name of the missing list is issued.

▶ SAVASPDEV

Specifies the name of the auxiliary storage pool (ASP) device from which private authority information was saved. The private authority information is restored for later use by the Restore Authority (RSTAUT) function.

***ANY:** The private authority information saved from all ASPs included in the save operation is restored.

***:** The private authority information saved from the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group is restored.

***SYSBAS:** The private authority information saved from the system ASP and all basic user ASPs is restored.

***CURASPGRP:** If the current thread has an ASP group, the private authority information saved from all independent ASPs in the ASP group is restored.

auxiliary-storage-pool-device-name: Specify the name of the independent ASP from which the private authority information was saved. <<

OMITUSRPRF

Specifies the names of one or more user profiles to be omitted from the restore.

***NONE:** None of the user profiles is omitted from the restore.

generic-user-profile-name:* Specify one or more generic names of sets of user profiles to be omitted from the restore. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. For more information on the use of generic functions, refer to generic names.

user-profile-name: Specify one or more names of specific user profiles that are to be omitted from the restore. Both generic names and specific names can be specified in the same command.

SECDDTA

Specifies whether all authority information or only the private authorities are restored for the specified user profiles and auxiliary storage. Also specifies whether the password and group linkages are to be restored for the specified user profiles.

***USRPRF:** All of the specified user profiles, authorization lists, authority holders, and private authorities, saved by the SAVSYS or SAVSECDDTA commands are restored.

***PVTAUT:** Only the private authorities for the specified user profiles and auxiliary storage pools are restored. The information is used by the Restore Authority (RSTAUT) command to restore the private authorities to the referenced objects.

***PWDGRP:** The passwords and group linkages for the specified user profiles are restored with the user profiles.

>> ***DCM:** Only the information required by Digital Certificate Manager (DCM) is restored.

Note: *PWDGRP cannot be specified if USRPRF(*USRPRF) is also specified.

Note: *PVTAUT cannot be specified if USRPRF(*NEW) is also specified.

Note: *DCM is only supported for save media that is at release V5R1M0 or later.



OUTPUT

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

***NONE:** No output is created.

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

Note:

You must specify the database file name on the OUTFILE parameter when OUTPUT(*OUTFILE) is specified.

OUTFILE

Specifies the qualified name of the database file to which the information about the object is directed when *OUTFILE is specified on the OUTPUT parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QASRRSTO in QSYS with the format name QSRRST as a model.

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

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

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

library-name: Specify the name of the library to be searched.

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

OUTMBR

Specifies the name of the database file member to which the output is directed. If a member already exists, the system uses the second element of this parameter to determine whether the member is cleared before the new records are added. If the member does not exist and a member name is not specified, the system creates a member with the name of the output file specified on the OUTFILE parameter. If an output file member name is specified, but the member does not exist, the system creates it.

Element 1: Member to Receive Output

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

member-name: Specify the name of the file member that receives the output. If OUTMBR(*member-name*) is specified and the member does not exist, the system creates it. If the member exists, the user can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Operation to Perform on Member

***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.

OPTFILE

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume. For more information on specifying path names, refer to path names.

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

'optical-directory-path-name/':* The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

Examples for RSTUSRPRF

Example 1: Restoring All Profiles

```
RSTUSRPRF DEV(TAP01) SEQNBR(*SEARCH) ENDOPT(*REWIND)
```

The saved version of all user profiles contained on the tape currently put on the tape drive named TAP01 is restored to the system. The tape is searched for the file, and the tape is rewound on completion or at the end of restore.

Example 2: Restoring Specific User Profiles

```
RSTUSRPRF DEV(TAP01) USRPRF(USRA USRB USRC USER*)
```

The saved version of all the user profiles must exist on the tape placed on tape drive TAP01. User profiles USRA, USRB, and USRC, along with all the user profiles whose names start with USER, are restored to the system.

Example 3: Restoring User Profiles from a Save File

```
RSTUSRPRF DEV(*SAVF) USRPRF(USRX USRY)
SAVF(QGPL/SAVESEC)
```

This command restores user profiles USRX and USRY to the system from the save file SAVESEC in library QGPL.

Example 4: Reporting Information about User Profiles Restored and Not Restored

```
RSTUSRPRF DEV(TAP01) USRPRF(*ALL) OUTPUT(*OUTFILE)
OUTFILE(PRFS92) OUTMBR(FOURQT *ADD)
```

This command restores all user profiles from the tape device TAP01. A list reporting information about user profiles restored and not restored is directed to the output file PRFS92. The output is received in the member FOURQT as an addition to existing information in the member.

Error messages for RSTUSRPRF

***ESCAPE Messages**

CPD3774

USRPRF(*ALL) required when MAIL(*YES) specified.

CPF2206

User needs authority to do requested function on object.

CPF370C

Not authorized to ALWOBJDIF parameter.

CPF3709

Tape devices do not support same densities.

- CPF3727**
Duplicate device &1 specified on device name list.
- CPF3728**
Device &1 specified with other devices.
- CPF3733**
&2 &1 in &3 previously damaged.
- CPF3738**
Device &1 used for save or restore is damaged.
- CPF3743**
File cannot be restored, displayed, or listed.
- CPF3748**
Object information for library &1 damaged.
- CPF376B**
File &1 not found.
- CPF3767**
Device &1 not found.
- CPF3768**
Device &1 not valid for command.
- CPF3775**
Not all user profiles or authority objects restored.
- CPF3780**
Specified file for library &1 not found.
- CPF3782**
File &1 in &2 not a save file.
- CPF3785**
Not all subsystems ended.
- CPF3793**
Machine storage limit reached.
- CPF3794**
Save or restore operation ended unsuccessfully.
- CPF3796**
Storage limit exceeded for user profile &4.
- CPF380C**
Library &1 not restored.
- CPF3812**
Save file &1 in &2 in use.
- CPF908A**
Requester &1 not enrolled.
- CPF9812**
File &1 in library &2 not found.
- » **CPF9814**
Device &1 not found. «
- » **CPF9833**
*CURASGRP or *ASGRP PRI specified and thread has no ASP group. «

RSMBKP (Resume Breakpoint) Command Description

RSMBKP Command syntax diagram

Purpose

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

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

Restriction: This command is valid only for programs in the debug mode and only when the program is stopped at a user-defined breakpoint. That is, this command is not valid at a breakpoint caused by an unmonitored message. To start the debug mode, use the STRDBG (Start Debug) command.

There are no parameters for this command.

Example for RSMBKP

RSMBKP

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

Error messages for RSMBKP

*ESCAPE Messages

CPF1999

Errors occurred on command.

RSMCTLR CY (Resume Controller Recovery) Command Description

RSMCTLR CY Command syntax diagram

Purpose

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

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

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

Required Parameter

CTL Specifies the controller whose recovery is started again. Valid types of controllers are:

CTL Value

Controller

5251 Display station

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

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

***BSCT**

BSC device (Multipoint tributary and 3270 Device Emulation)

***APPC**

Advanced program-to-program communications

***WSC** Local work station

***WSCE**

Local work station (extended)

Specify the name of the controller (as specified in the controller description) for which error recovery procedures are to resume.

Example for RSMCTLR CY

```
RSMCTLR CY CTL(TROLL3)
```

This command resumes error recovery procedures for the controller TROLL3.

Error messages for RSMCTLR CY

*ESCAPE Messages

CPF2703

Controller description &1 not found.

CPF5924

Controller &1 does not allow automatic error recovery.

CPF5928

Controller &1 not varied on.

CPF5929

Controller &1 assigned to another job.

CPF5930

Recovery not resumed for controller &1.

CPF5931

Recovery not resumed for controller &1.

CPF5935

Error occurred during command processing.

CPF5936

Not authorized to controller &1.

RSMDEVRCY (Resume Device Recovery) Command Description

RSMDEVRCY Command syntax diagram

Purpose

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

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

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

Required Parameter

DEV Specifies the device whose recovery is started again. Valid types of devices are:

DEV Type

Device

5219 Printer (work station)

5224 Printer (work station)

5225 Printer (work station)

5251 Display station

5252 Dual display station

5256 Printer (work station)

5291 Display station

5292 Display station

***PLU1** Physical unit (type 1)

***BSC** BSC device

***BSCT**

BSC multipoint tributary

***APPC**

Advanced program-to-program communications

Specify the name of the device (as specified in the device description) for which error recovery procedures are to resume.

Example for RSMDEVRCY

```
RSMDEVRCY DEV(WSPR03)
```

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

RSMHTTPCRL (Resume HTTP Crawl) Command Description

Note: To use this command, you must have the 5722-DG1 (HTTP Server) product installed.

Purpose

The Resume HTTP (RSMHTTPCRL) command starts a paused crawling session.

Restriction: You must have *IOSYSCFG special authority to use this command.

Parameters

DOCLIST

Specifies the document list file where the list of path names are stored.

Error messages for RSMHTTPCRL

*ESCAPE messages

HTP165D

Request to resume a crawling session failed. Reason &1.

RSMLINRCY (Resume Line Recovery) Command Description

RSMLINRCY Command syntax diagram

Purpose

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

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

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

Required Parameter

LINE Specifies the line whose recovery is started again.

Example for RSMLINRCY

```
RSMLINRCY LINE(NYC2)
```

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

Error messages for RSMLINRCY

*ESCAPE Messages

CPF2704

Line description &1 not found.

CPF5917

Not authorized to line description &1.

CPF5932

Cannot access line &1.

CPF5933

Line &1 not varied on.

CPF5934

Recovery not resumed for line &1.

CPF5935

Error occurred during command processing.

RSMNWIRCY (Resume Network Interface Recovery) Command Description

RSMNWIRCY Command syntax diagram

Purpose

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

Required Parameter

NWID Specifies the name of the network interface description whose automatic error recovery is restarted.

Example for RSMNWIRCY

```
RSMNWIRCY  NWID(ISDNNET)
```

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

Error messages for RSMNWIRCY

***ESCAPE Messages**

CPF591A

Not authorized to network interface description &1.

CPF593A

Network interface &1 not varied on.

CPF593B

Network interface description &1 not found.

CPF593C

Cannot access network interface &1.

CPF593D

Recovery not resumed for network interface &1.

RSMRTVBRM (Resume Retrieve Using BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.

RSMRTVBRM Command syntax diagram

Purpose

You can resume delayed retrieve operations for objects, including those held due to lack of authority or held due to lack of disk space. The Resume Retrieve using BRM (RSMRTVBRM) command attempts to retrieve objects from tape.

Example for RSMRTVBRM

Example 1: Resuming a Retrieve for Security Items

```
RSMRTVBRM HDLOBJ(*SECURITY)
```

In the example, the retrieve operation is resumed for items that were held due to lack of authorization.

Error messages for RSMRTVBRM

None

RTVAUTLE (Retrieve Authorization List Entry) Command Description

RTVAUTLE Command syntax diagram

Purpose

The Retrieve Authorization List Entry (RTVAUTLE) command is used in a CL program or REXX procedure to retrieve the authorities that a user has on the authorization list. It can be used with the Change Authorization List Entry command to change the user's authorities to include new authorities in addition to the existing authorities for the user.

The authorization list name and user name must be specified. The variables for each of the authorities the user might have are returned blank if the user does not have the authority; they are returned with the correct value for the Change Authorization List Entry (CHGAUTLE) command if the user has the authority. The values are returned in the specified variables for the specified user.

Users who have *AUTLMGT authority or who own the authorization list can retrieve authority for any user on the list. Other users can retrieve their own authorities or the authority of *PUBLIC.

The following values are returned if the user has the authority specified by the keyword:

Keyword	Value
ADD	*ADD
ALL	*ALL
AUTLMGT	*AUTLMGT
CHANGE	*CHANGE
DELETE	*DLT
EXCLUDE	*EXCLUDE
EXECUTE	*EXECUTE
OBJALTER	*OBJALTER
OBJEXIST	*OBJEXIST

Keyword	Value
OBJMGT	*OBJMGT
OBJOPR	*OBJOPR
OBJREF	*OBJREF
READ	*READ
UPDATE	*UPD
USE	*USE

The CL prompt for this command lists the minimum length for the variables next to the appropriate parameters to be retrieved. For character variables, a single number is shown.

Required Parameters

AUTL Specifies the name of the authorization list that the user's authorities come from.

USER Specifies the name of the user whose information is being retrieved. If a variable is specified, it must be 10 characters in length and contain a user name or the value *PUBLIC.

***PUBLIC:** The information returned in the specified parameter is for the users who do not have any specific authority to the authorization list, and whose user groups do not have any specific authority to the authorization list.

user profile name: Specify the user profile name of the user whose information is being retrieved.

Optional Parameters

ALL Specifies the name of a variable that is used to return *ALL if the user has *ALL authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *ALL authority.

CHANGE

Specifies the name of a variable that is used to return *CHANGE if the user has *CHANGE authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *CHANGE authority.

USE Specifies the name of a variable that is used to return *USE if the user has *USE authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *USE authority.

EXCLUDE

Specifies the name of a variable that is used to return *EXCLUDE if the user has *EXCLUDE authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *EXCLUDE authority.

OBJALTER

Specifies the name of a variable that is used to return *OBJALTER if the user has *OBJALTER authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *OBJALTER authority.

OBJEXIST

Specifies the name of a variable that is used to return *OBJEXIST if the user has *OBJEXIST authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *OBJEXIST authority.

OBJMGT

Specifies the name of a variable that is used to return *OBJMGT if the user has *OBJMGT authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *OBJMGT authority.

OBJOPR

Specifies the name of a variable that is used to return *OBJOPR if the user has *OBJOPR

authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *OBJOPR authority.

OBJREF

Specifies the name of a variable that is used to return *OBJREF if the user has *OBJREF authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *OBJREF authority.

READ Specifies the name of a variable that is used to return *READ if the user has *READ authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *READ authority.

ADD Specifies the name of a variable that is used to return *ADD if the user has *ADD authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *ADD authority.

UPDATE

Specifies the name of a variable that is used to return *UPD if the user has *UPD authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *UPD authority.

DELETE

Specifies the name of a variable that is used to return *DLT if the user has *DLT authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *DLT authority.

EXECUTE

Specifies the name of a variable that is used to return *EXECUTE if the user has *EXECUTE authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *EXECUTE authority.

AUTLMGT

Specifies the name of a variable that is used to return *AUTLMGT if the user has *AUTLMGT authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *AUTLMGT authority.

Example for ADDAUTLE

```
ADDAUTLE AUTL(PAYROLL) USER(TOM)
AUT(*OBJOPR *READ *UPD *AUTLMGT)
```

When user Smith calls a CL program containing the following:

```
DCL   &CHG           *CHAR 10
DCL   &ALL           *CHAR 10
DCL   &USE           *CHAR 10
DCL   &EXCL         *CHAR 10
DCL   &OBJJOP       *CHAR 10
DCL   &ALTER        *CHAR 10
DCL   &REFER        *CHAR 10
DCL   &READ         *CHAR 10
DCL   &ADD          *CHAR 10
DCL   &UPD          *CHAR 10
DCL   &DLT          *CHAR 10
DCL   &EXEC         *CHAR 10
DCL   &AUTLM        *CHAR 10
```

```
RTVAUTLE AUTL(PAYROLL) USER(TOM) USE(&USE)
OBJOPR(&OBJJOP) AUTLMGT(&AUTLM)
```

This command retrieves the following authorities from the authorization list PAYROLL for user TOM: *USE, *OBJOPR, and *AUTLMGT. If TOM does not have the authority, blanks are returned.

Error messages for RTVAUTLE

***ESCAPE Messages**

CPF22A7

User &1 not on authorization list &2, no authorities retrieved.

CPF22A8

Not authorized to retrieve authorities for user &1.

CPF2204

User profile &1 not found.

CPF2283

Authorization list &1 does not exist.

CPF2289

Unable to allocate authorization list &1.

RTVBCKUP (Retrieve Backup) Command Description

RTVBCKUP Command syntax diagram

Purpose

The Retrieve Backup (RTVBCKUP) command is used in a CL program or REXX procedure to retrieve the options in one of the predefined backups into CL variables. More information on backup is in the Backup, Recovery, and Availability topic in the Information Center.

Required Parameter

BCKUOPT

Specifies the backup options to be retrieved.

***DAILY:** The daily backup options are retrieved.

***WEEKLY:** The weekly backup options are retrieved.

***MONTHLY:** The monthly backup options are retrieved.

Optional Parameters

DEV Specifies the name of the CL variable that receives the device value. The variable has a minimum length of 43 characters. The value returned is a character string of four 10-character device names, separated by blanks.

TAPSET

Specifies the name of the CL variable that receives the tape set names. The variable has a minimum length of 34 characters (seven 4-character tape set names, separated by blanks).

CLRTAP

Specifies the name of the CL variable that receives the indicator for clearing the tape for backup. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

SBMJOB

Specifies the name of the CL variable that receives the indicator of whether the backup is run as a batch job. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

CHGONLY

Specifies the name of the CL variable that receives the indicator for saving changed objects only. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

PRTRPT

Specifies the name of the CL variable that receives the indicator for printing a report of saved objects. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

LIB Specifies the name of the CL variable that receives the value specifying the libraries to save with this backup. The variable must have a minimum length of 10 characters. A value of *ALLUSR, *FROMLIST, or *NONE is returned.

FLR Specifies the name of the CL variable that receives the value specifying the folders to save with this backup. The variable must have a minimum length of 10 characters. A value of *ALL, *FROMLIST, or *NONE is returned.

DIR Specifies the name of the CL variable that receives the value specifying the user directories to save with this backup. The variable must have a minimum length of 10 characters. A value of *ALLUSR or *NONE is returned.

SECDDTA

Specifies the name of the CL variable that receives the indicator for saving security data. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

CFG Specifies the name of the CL variable that receives the indicator for saving configuration data. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

MAIL Specifies the name of the CL variable that receives the indicator for saving OfficeVision mail. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

CAL Specifies the name of the CL variable that receives the indicator for saving OfficeVision calendars. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

EXITPGM

Specifies the name of the CL variable that receives the name of the user program to call before and after the backup is run. The variable must have a minimum length of 10 characters. If no exit program is specified, *NONE is returned.

EXITPGMLIB

Specifies the name of the CL variable that receives the name of the library that contains the exit program. The variable must have a minimum length of 10 characters. If no exit program is specified, blanks are returned. If *LIBL is returned, the program uses the library list.

Example for RTVBCKUP

```
RTVBCKUP BCKUOPT(*DAILY) SBMJOB(&SBMJOBVAR)
        LIB(&LIBVAR)
```

This command retrieves the SBMJOB and LIB values for the daily backup into the CL variables SBMJOBVAR and LIBVAR respectively.

Error messages for RTVBCKUP***ESCAPE Messages****CPF1EE3**

Not authorized to backup options.

CPF1E6C

Backup options in use.

CPF1E67

Backup options and library backup list damaged.

CPF1E99

Unexpected error occurred.

RTVBNSRC (Retrieve Binder Source) Command Description

RTVBNSRC Command syntax diagram

Purpose

The Retrieve Binder Source (RTVBNSRC) command can be used to retrieve the exports from **>>** a set of module objects or a service program or both, **<<** and place them (along with the binder language statements needed for the exports) in a specified file member. This file member can later be used as input to the SRCMBR parameter on the Create Service Program (CRTSRVPGM) or the Update Service Program (UPDSRVPGM) commands. After the binder language has been retrieved into a source file member, you can edit the binder language to make changes as needed.

The CRTSRVPGM command uses a binder language file specified on the SRCFILE parameter to identify the exports from the service program. The RTVBNSRC command can be useful in helping you automatically create this binder language.

Restrictions:

1. You must have *USE authority to the Create Source Physical File (CRTSRCPF) command if the file does not exist.
2. You must have *USE authority to the Reorganize Physical File Member (RGZPFM) command.
3. You must have *USE authority to the Add Physical File Member (ADDPFM) command if the member does not exist.
4. You must have *USE authority to the modules **>>** or service program **<<** from which the exports are being retrieved.
5. You must have *EXECUTE authority to the libraries in which the modules **>>** or service program **<<** exist.
6. If the source file and member to receive the binder language exist, you must have *CHANGE and *OBJALTER or *OBJMGT authority to the file and *EXECUTE authority to the library that contains the file.
7. If the source file exists but the source member needs to be created, you must have *OBJOPR, *OBJMGT, and *ADD authority to the file and *EXECUTE, *READ and *ADD authority to the library that contains the file.
8. If the source file and member need to be created, you must have *EXECUTE, *READ and *ADD authority to the library.

Optional Parameters

MODULE

Specifies the list of modules from which to retrieve the exported symbols. If duplicate module and library specifications are found, only the first instance of the duplicate module and library is used.

>> At least one value must be specified for this parameter or the SRVPGM parameter. **<<**

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

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

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

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

library-name: Specify the name of the library to be searched.

***ALL:** The exported symbols from all of the modules in the specified library are retrieved.

generic-module-name:* Specify a generic module name from which to retrieve the exported symbols. All modules that have names with the same prefix in the specified library or libraries are used. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. For more information on the use of generic names, refer to generic names.

module-name: Specify the name of the module from which to retrieve the exported symbols.



SRVPGM

Specifies the service program from which to retrieve the exported symbols. At least one value must be specified for this parameter or the MODULE parameter.

The name of the service program can be qualified by one of the following library values:

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

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

library-name: Specify the name of the library to be searched.

service-program-name: The name of the service program from which to retrieve the exported symbols.



SRCFILE

Specifies the name of the source file that is to hold the binder language for the exported symbols. If the source file does not exist, it is created.

Notes:

1. Only physical files of type *SRC can be specified. Distributed data management (DDM) files are not supported.
2. If the source file to receive the binder language exists, its record length must be a minimum of 92 bytes.

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

***LIBL:** All libraries in the job's library list are searched until the first match is found. If a source file by the name specified is not found in the library list, it is created in the current library. If there is no current library, the QGPL library is used.

***CURLIB:** The current library for the job is searched. If a source file by the name specified does not exist, it is created in the current library. If there is no current library, the QGPL library is used.

library-name: Specify the name of the library to be searched. If a source file by the name specified is not found in this library, the source file is created in this library.

QSRVSRC: The source file name is the default source file name, QSRVSRC.

source-file-name: Specify the name of the source file.

SRCMBR

Specifies the name of the member in the source file that is to hold the binder language for the exported symbols. Only one source member will contain the binder language, even if the binder language is generated from exported symbols retrieved from multiple modules.

If the member does not exist in the source file specified, the member is created.

***DFT:** ➤ The name of the source file member is taken from the value specified on the SRVPGM parameter if a service program is specified. Otherwise, ⚡ the name of the source file member is taken from the value specified on the MODULE parameter.

- If only one module is specified, the name of that module is the member used.
- If more than one module is specified, the name of the first module specified is used.
- If the value *ALL or a generic name is specified, the module name of the first occurrence found is the source member name used.

source-file-member-name: Specify the name of the member that contains the binder language source.

MBROPT

Specifies whether the generated binder language statements are replaced or added to the existing statements.

***REPLACE:** The system clears the existing member and adds the new records.

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

Note:

If the member already exists and, for example, already contains STRPGMEXP and ENDPGMEXP statements, the member may contain multiple STRPGMEXP and ENDPGMEXP statements in the binder language at the end of this operation. You must edit these multiple statements in order to use the binder language for the CRTSRVPGM command.

Examples for RTVBNSRC

```
RTVBNSRC  MODULE(MYLIB/*ALL)
          SRCFILE(MYLIB/MYBINDFILE)  MBROPT(*ADD)
```

This command retrieves the exports from all modules in the library MYLIB, and places them in the source member with the name of the first module found. If this source member does not exist in the file MYBINDFILE in the library MYLIB, it is created. The export statements are added to the end of the

member. If multiple start and end program export statements exist in the file when this command is ended, the source member must be edited before it is used to create a service program. Either the extra STRPGMEXP, ENDPGMEXP statements can be removed, or the PGMLVL parameter can be added to the STRPGMEXP statements, if some of the export blocks are for previous versions of the service program.

Error messages for RTVBNDSRC

*ESCAPE Messages

CPF5CA5

Record length too small for data base source file.

CPF5D06

Not authorized to library &2, or file &1 in library &2, or member &3.

CPF5D08

File &1 in library &2 not supported file type.

CPF9810

Library &1 not found.

CPF9832

Function not supported for DDM file &2.

CPF9834

Input file &1 in library &2 is not a source file.

CPF9848

Cannot open file &1 in library &2 member &3.

CPF9899

Error occurred during processing of command.

RTVCLSRC (Retrieve CL Source) Command Description

RTVCLSRC Command syntax diagram

Purpose

The Retrieve CL Source (RTVCLSRC) command is used to retrieve the source statements from a CL program used to compile that program. These source statements are placed into a source file member, which can be used as input when recompiling the CL program.

Required Parameters

PGM Specifies the qualified name of the CL program whose source statements are being retrieved.

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

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

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

library-name: Specify the name of the library to be searched.

program-name: Specify the name of the CL program whose source statements are being retrieved.

SRCFILE

Specifies the qualified name of the previously created database source file into which the CL source statements are being written.

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

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

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

library-name: Specify the name of the library to be searched.

source-file-name: Specify the name of the database source file that contains the CL source statements.

Optional Parameters

SRCMBR

Specifies the name of the database source file member into which the CL source statements are being written. If not specified, the CL program name is assumed. If the member existed before running the command, it is cleared before any source statements are written into it. If the member did not exist, it is created.

***PGM:** The name of the CL program is used as the member name.

source-member-name: Specify the name of the source file member that contains the CL source statements.

Example for RTVCLSRC

```
RTVCLSRC PGM(JOHN1/TEXT1) SRCFILE(JOHN2)
SRCMBR(JOHN3)
```

This command retrieves the source statements from the CL program named TEXT1 in library JOHN1. The retrieved source statements are placed into the file named JOHN2, and are named as member JOHN3.

Error messages for RTVCLSRC

*ESCAPE Messages

CPF0560

Program &1 in &2 not a CL program.

CPF0561

Unable to retrieve CL source from CL program &2.

CPF0562

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

CPF0563

Record length too small for data base source file.

CPF0564

Unable to add data base member &3 to file.

- CPF0565**
Source from CL program &4 not retrieved.
- CPF0566**
Source not available for CL program &1 in &2.
- CPF9801**
Object &2 in library &3 not found.
- CPF9803**
Cannot allocate object &2 in library &3.
- CPF9805**
Object &2 in library &3 destroyed.
- CPF9806**
Cannot perform function for object &2 in library &3.
- CPF9807**
One or more libraries in library list deleted.
- CPF9808**
Cannot allocate one or more libraries on library list.
- CPF9809**
Library &1 cannot be accessed.
- CPF9810**
Library &1 not found.
- CPF9811**
Program &1 in library &2 not found.
- CPF9820**
Not authorized to use library &1.
- CPF9821**
Not authorized to program &1 in library &2.
- CPF9822**
Not authorized to file &1 in library &2.
- CPF9830**
Cannot assign library &1.
- CPF9848**
Cannot open file &1 in library &2 member &3.
- CPF9849**
Error while processing file &1 in library &2 member &3.

RTVCLNUP (Retrieve Cleanup) Command Description

RTVCLNUP Command syntax diagram

Purpose

The Retrieve Cleanup (RTVCLNUP) command is used in a CL program or REXX procedure to retrieve a cleanup operation value. The value is returned (copied) to the specified CL variable.

Optional Parameters

ALWCLNUP

Specifies the name of the CL variable that receives the allow cleanup value. The variable named has a minimum length of 4 characters. *YES is returned if the cleanup operation is allowed to run. Otherwise, *NO is returned.

STRTIME

Specifies the name of the CL variable that receives the time the cleanup operation starts each day. The variable named has a minimum length of 10 characters. The special value *NONE or *SCDPWROFF, or the start time is returned.

USRMSG

Specifies the name of the CL variable that receives the value for deleting user messages on user profile message queues. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days user messages are kept before they are deleted is returned.

SYSMMSG

Specifies the name of the CL variable that receives the value for deleting messages on the QSYSOPR message queue and on work station message queues. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days system messages are kept before they are deleted is returned.

SYSPRT

Specifies the name of the CL variable that receives the value for deleting job logs and other system output. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days job logs are kept before they are deleted is returned.

SYSLOG

Specifies the name of the CL variable that receives the value for deleting system journals, history files, problem log files, alert database, and program temporary fixes. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days system journals and system logs are kept before they are deleted is returned.

CALITM

Specifies the name of the CL variable that receives the value for deleting OfficeVision calendar items. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days calendar items are kept before they are deleted is returned.

JOBQ Specifies the name of the CL variable that receives the name of the job queue to which the cleanup batch jobs are submitted. The variable named has a minimum length of 10 characters. The name of the job queue under which cleanup batch jobs are run is returned.

JOBQLIB

Specifies the name of the CL variable that receives the library name of the job queue to which the cleanup batch jobs are submitted. The variable named has a minimum length of 10 characters. The name of the library where the job queue is located is returned.

RUNPTY

Specifies the name of the CL variable that receives the 2-digit value, ranging from 1 through 99, that is the run (or processing) priority for routing steps that are part of the job. For additional information on run priority, refer to this parameter description in the CHGJOB (Change Job) command. The variable must be a 2-digit decimal variable specified with no decimal positions.

JRNRCVSIZ

Specifies the name of the CL variable that receives the value for the journal receiver size ranging from 1 through 1,919,999 (KB). When the size of the space for the journal receiver is larger than the size specified by this value, Operational Assistant (OA) automatic cleanup function will detach it. The variable must be an 7-digit decimal variable specified with no decimal positions.

Examples for RTVCLNUP**Example 1: Retrieving Number of Days Messages are Kept**

```
DCL VAR(&UMSGDAYS) TYPE(*CHAR) LEN(5)
RTVCLNUP USRMSG(&UMSGDAYS)
```

These commands retrieve the number of days that user messages are kept before being deleted.

Example 2: Retrieving Time Cleanup Operation Starts

```
DCL VAR(&CLNUPTIME) TYPE(CHAR) LEN(10)
RTVCLNUP STRTIME(&CLNUPTIME)
```

These commands retrieve the time that the cleanup operation starts.

Example 3: Retrieving Run Priority

```
DCL VAR(&RPTY) TYPE(*DEC) LEN(2 0)
RTVCLNUP RUNPTY(&RPTY)
```

This command retrieves the run priority for the cleanup job. The run priority number is copied into the CL variable &RPTY. The variable must be a 2-digit decimal variable with no decimal positions.

Example 4: Retrieving Journal Receiver Size

```
DCL VAR(&JRNSIZ) TYPE(*DEC) LEN(7 0)
RTVCLNUP JRNRCVSIZ(&JRNSIZ)
```

This command retrieves the journal receiver size for the cleanup job. The journal receiver size is copied into the CL variable &JRNSIZ. The variable must be a 7-digit decimal variable with no decimal positions.

Error messages for RTVCLNUP

*ESCAPE Messages

CPF1E2B

Power scheduler and cleanup options not found.

CPF1E33

Cleanup options or power schedule in use by another user.

CPF1E99

Unexpected error occurred.

RTVCFGSRC (Retrieve Configuration Source) Command Description

RTVCFGSRC Command syntax diagram

Purpose

The Retrieve Configuration Source (RTVCFGSRC) command retrieves CL command source to create existing configuration objects. This command allows easy portability of configuration objects from one system to another and provides a method of saving configurations without having to run the SAVSYS command.

Required Parameters

CFGD Specifies the configuration objects of the CL source being retrieved. Multiple names can be specified; generic names are accepted. The CFGTYPE parameter specifies the type of object to which the names refer.

***ALL:** All objects specified by the CFGTYPE parameter are retrieved.

generic-configuration-description-name:* Specify the generic name of the configuration description name. A generic name is a character string of one or more characters followed by an asterisk (*);

for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. For more information on the use of generic names, refer to generic names.

configuration-description-name: Specify the user-defined name of the configuration description.

CFGTYPE

Specifies the object type to which the names in the CFGD parameter are referring.

***ALL**: All network servers, network interfaces, lines, controllers, devices, connection lists, modes, classes-of-service, NetBIOS, and Internet Packet Exchange descriptions matching the specified names are retrieved in the following order:

1. Connection Lists
2. Network server descriptions
3. Network Interfaces
4. Non-TDLC line descriptions
5. Non-TDLC controller descriptions
6. TDLC line descriptions
7. TDLC controller descriptions
8. Device descriptions
9. Mode descriptions
10. Class-of-service descriptions
11. NetBIOS descriptions
12. SWTCTLLST for line descriptions
13. SWTLINLST for controller descriptions
14. SWTNWILST for line descriptions
15. Printer for remote displays

***NWSD**: All network server descriptions matching the specified names are retrieved.

***NWID**: All network interfaces matching the specified names are retrieved.

***LIND**: All lines matching the specified names are retrieved.

***CTLD**: All controllers matching the specified names are retrieved.

***DEV**: All devices matching the specified names are retrieved.

***MOD**: All modes matching the specified names are retrieved.

***COS**: All classes-of-service matching the specified names are retrieved.

***CN**: All connection lists matching the specified names are retrieved.

***NTB**: All NetBIOS descriptions matching the specified names are retrieved.

Optional Parameters

SRCFILE

Specifies the name of the previously created database source file where the retrieved source is written.

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

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

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

library-name: Specify the name of the library to be searched.

QCLSRC: The retrieved source is written in the QCLSRC source file.

source-file-name: Specify the name of the source file where the retrieved source is being written.

SRCMBR

Specifies the member name in the source file where the retrieved source is written. If the member does not exist, it is created.

***CFGD:** If *ALL, a generic name or list of names is specified on the CFGD parameter, the name of the source file member is CFGSRC. Otherwise, the name specified on the CFGD parameter is used as the member name.

source-member-name: Specify the source file member name.

RTVOPT

Specifies which attachment information is retrieved for the specified objects.

This parameter cannot be specified if CFGTYPE(*ALL) is specified. If CFGTYPE(*ALL) is specified, the source is retrieved for objects of all types with names as specified on the CFGD parameter, followed by switched attachment information (SWTCTLLST, SWTNWILST, and SWTLINLST parameters on Change Line, Change Network Interface, and Change Controller commands).

***NET:** The source for the configuration object names specified on the CFGD parameter, of the type specified on the CFGTYPE parameter are retrieved. Also retrieved are the downward-attached configuration objects. For example, if CFGTYPE(*CTLD) is specified, the configuration source for devices attached to the specified controllers is also retrieved. For CFGTYPE(*CTLD), the switched attachment configuration information (if any) is retrieved in the form of Change Controller (CHGCTLxxx) commands including the SWTLINLST parameter following the Create Controller (CRTCTLxxx) commands.

***OBJ:** The source for the configuration object names specified on the CFGD parameter, of the type specified on the CFGTYPE parameter, is retrieved.

MBROPT

Specifies whether the new records replace or are added to the existing records.

***REPLACE:** The system clears the existing member and adds the new records.

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

TEXT

Specifies the text that briefly describes the configuration description. More information on this parameter is in Commonly used parameters.

***CFGDTXT:** If *ALL is specified on the CFGD parameter or a generic name or list of names, then *BLANK is used as the text. If MBROPT(*ADD) is specified, the member's text is replaced; otherwise, the text of the configuration object being retrieved is used as the text.

***BLANK:** Text is not specified.

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

Example for RTVCFGSRC

```
RTVCFGSRC CFGD(CTL*) CFGTYPE(*CTLD)
          SRCMBR(CTLS) RTVOPT(*OBJ)
```

This command places CL source statements in the file member CTLS in the source file QCLSRC. These source statements can be used to re-create object descriptions for all existing controllers with names beginning with CTL.

Error messages for RTVCFGSRC

*ESCAPE Messages

CPF2207

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

CPF263E

File member &1 contains its maximum sequence number.

CPF263F

No objects found.

CPF264A

Record length of &1 in library &2 is too small.

CPF264C

Source file member &1 not found.

CPF9810

Library &1 not found.

CPF9820

Not authorized to use library &1.

CPF9847

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

CPF9848

Cannot open file &1 in library &2 member &3.

CPF9849


Error while processing file &1 in library &2 member &3.

RTVCFGSTS (Retrieve Configuration Status) Command Description

RTVCFGSTS Command syntax diagram

Purpose

The Retrieve Configuration Status (RTVCFGSTS) command is used in a CL program or REXX procedure to retrieve the status of a configuration object.

More information is in the Communications Management  book and in the Communications

Configuration  book.

Required Parameters

CFGD Specifies the name of the configuration object whose status is retrieved. Valid values range from 1 through 10 characters.

CFGTYPE

Specifies the type of configuration object.

***NWS:** The object is a network server description.

***NWI:** The object is a network interface description.

***LIN:** The object is a line description.

***CTL:** The object is a controller description.


***DEV:** The object is a device description.

STSCDE

Specifies the name of the variable that will contain the retrieved status. In a CL program, the variable name should be a decimal variable of length (5 0).

The possible values which can be returned include:

Value	Definition
-------	------------

- | | |
|----|--|
| 0 | VARIED OFF -The system is not using the description. |
| 10 | VARIED OFF PENDING -The description is being varied off. During this time the system may be taking down the tasks which managed the resource. |
| 20 | VARIED ON PENDING -The description is being varied on. During this time the system may be putting tasks in place to manage the resource, downloading microcode to an I/O processor, communicating with data circuit-terminating equipment (DCE), and so on. |
| 30 | VARIED ON -The tasks that manage the network interface, network server, line, controller, or device have been put in place by the system, and the system has the capability to communicate with it. |
| 32 | VARIED ON/CNN PENDING - The first of a pair of OptiConnect controllers is varied on but its attached device is not yet in a varied on state. |
| 40 | CONNECT PENDING -This status is only valid for switched IDLC, SDLC, BSC, or asynchronous lines. The line is in this status while waiting for the switched connection to be established; this connection can be either a dial or an answer. |
| 50 | SIGNON DISPLAY -This status is only valid for display devices. The system is preparing the device to receive the sign-on display, sending the sign-on display, or the actual sign-on display is at the display station. |
| 51 | ACTIVE/CNN PENDING - The first of a pair of OptiConnect controllers and its attached device are varied on and waiting for the OptiConnect path to be established. |
| 60 | ACTIVE -The object is successfully placed in VARIED ON status. For network interfaces and network servers, one or more attached lines are in VARY ON PENDING status or higher. For lines, one or more attached controllers is in a VARY ON PENDING status or higher. For controllers, one or more attached devices is in a VARY ON PENDING status or higher. For devices, active status varies depending on the type of device. More information is in the Communications Configuration  book. A display device which is at a second sign-on display as a result of pressing the system request key will be considered ACTIVE. |
| 63 | ACTIVE READER - The device is in use by a spool reader. |
| 66 | ACTIVE WRITER - The device is in use by a spool writer. |
| 70 | HELD -This status is only valid for Device Descriptions. The user or the system held the communications device to prevent it from communicating. The Release Communications Device (RLSCMNDEV) command can be used to release the device. |
| 80 | RCYPND -Error recovery is pending for the network interface, line, controller, or device. A message indicating what error occurred appears on the QSYSOPR message queue. |

- 90 **RCYCNL**-Error recovery is canceled for the network interface, line, controller, or device. An error occurred, and the operator replied with a C (to cancel error recovery) to a message, or the operator used a command (ENDNWIRCY, ENDLINRCY, ENDCTLR CY, ENDDDEVRCY) to end error recovery.
- 95 **SYSTEM REQUEST**- The display device has been requested by the system and its associated job has been suspended. This occurs as a result of a user pressing the System Request key.
- 100 **FAILED**- An error occurred for the network interface, network server, line, controller, or device that can be recovered only by varying off and on again.
- 103 **FAILED READER**- The device is in use by a spool reader and is not available.
- 106 **FAILED WRITER**- The device is in use by a spool writer and is not available.
- 107 **SHUTDOWN**- The NWSD was shut down using an AIX interface.
- 110 **DIAGNOSTIC MODE**-The network interface, network server, line, controller, or device resource is being used by problem analysis procedures to diagnose problems, and the resource cannot be used by other users.
- 111 **DAMAGED**-The network interface, network server, line, controller, or device description is damaged. This is a system error condition. Information indicating when this damage occurred appears in the history log (QHST). The description must be deleted and created once more before it can be used again.
- 112 **LOCKED**-The actual status of the resource cannot be determined because another job has an exclusive lock on the description. Retry at a later time, or use the Work with Object Lock (WRKOBJLCK) command to determine which job has the lock on the description.
- 113 **UNKNOWN**-The status indicator of the description cannot be determined. This is a system error condition. Use the Dump Object (DMPOBJ) command to dump the contents and/or attributes of the description to a spooled printer file, and contact an IBM representative.

Example for RTVCFGSTS

```
RTVCFGSTS CFGD(ND01) CFGTYPE(*LIN)
          STSCDE(&STSCODE)
```

This command retrieves the configuration status of the line configuration description ND01 for use in the CL variable &STSCODE.

Error messages for RTVCFGSTS

None

RTVCURDIR (Retrieve Current Directory) Command Description

RTVCURDIR Command syntax diagram

Purpose

The Retrieve Current Directory (RTVCURDIR) command is used in a CL program or REXX procedure to retrieve the name of the current directory into the specified CL variable. An absolute path name containing no symbolic links is retrieved. The length of the name of the current directory is also retrieved.

The CL prompt for this command lists the minimum length for retrieved variables next to the appropriate parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

1. *X authority is required to the current directory, and the user must have *RX authority to each directory in the path.
2. The maximum length of a directory name that can be retrieved is limited by the maximum length of a character variable.

Note:

The maximum length of a character variable cannot exceed 9999 bytes.

Required Parameters**RTNDIR**

Specifies the name of the CL variable that receives the name of the current directory. The variable must be a character variable. If the current directory name has fewer characters than the variable allows, the value is not padded.

DIRNAMLEN

Specifies the name of the CL variable that receives the length (in bytes) of the current directory name. This length can be longer than the length of the character variable to receive the directory name. The variable must be a 7-digit decimal variable specified with no decimal positions.

Example for RTVCURDIR

```
RTVCURDIR RTNDIR(&CD) DIRNAMLEN(&CDLEN)
```

This command retrieves the name of the current directory and the length of the name of the current directory.

Error messages for RTVCURDIR***ESCAPE Messages****CPFA085**

Home directory not found for user &1.

CPFA09C

Not authorized to object.

CPFA0A1

An input or output error occurred.

CPFA0A9


Object not found.

RTVDTAARA (Retrieve Data Area) Command Description


RTVDTAARA Command syntax diagram

Purpose

The Retrieve Data Area (RTVDTAARA) command is used in a CL program or REXX procedure to retrieve all or part of a specified data area and to copy it into a CL variable. RTVDTAARA does not retrieve any other attributes of the data area. Existence of the data area is not required at the time the CL program is compiled.

If the job is a group job, the data area specified may be the group data area (*GDA). This data area is automatically associated with the group, and it is inaccessible from jobs outside the group. The length of this character data area is 512 bytes. More information about group jobs is in the Work Management  book.

A local data area (*LDA) is a character data area that is 1024 bytes in length, and it is automatically associated with the job. Another job cannot access the local data area.

If the job is a prestart job, the data area specified may be the data area that contains program initialization parameter data (*PDA). This data area is automatically associated with the prestart job and is inaccessible from other jobs. The length of this character data area is 2000 bytes. More information about prestart jobs is in the Work Management  book.

When a local data area or group data area must be retrieved during the processing of the RTVDTAARA command, the data area is locked during the retrieval operation so that commands in other jobs cannot change or destroy it until the operation is complete. If the data area is shared with other jobs and is updated in steps involving more than one command in a job, the data area should be explicitly allocated to that job until all the steps have been performed. A data area other than a local data area, group data area, or program initialization parameter data area can be explicitly allocated with the Allocate Object (ALCOBJ) command. No allocation is necessary for a local data area, group data area, or program initialization parameter data area.

Restrictions:

1. To use this command, the user must have *USE authority for the data area and *EXECUTE authority for the library where the data area is located. No specific authority is required to retrieve the value of a local data area or group data area.
2. This command is conditionally threadsafe. The following restrictions apply:
 - a. Retrieving DDM data areas in a job that allows multiple threads is not threadsafe.
 - b. Retrieving DDM data areas will fail when more than one thread is active in a job.

Required Parameters

DTAARA

Specifies the qualified name of the data area whose value is retrieved.

The name of the data area can be qualified by one of the following library values:

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

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

library-name: Specify the name of the library to be searched.

Element 1: Data Area Name

data-area-name: Specify the name of the data area whose value is being retrieved.

Element 2: Data Area Options

***GDA:** The value of the group data area is being retrieved. This value is valid only if this is a group job.

***LDA:** The value of the local data area is being retrieved.

***PDA:** The value of the program initialization parameter data area is being retrieved. This value is valid only if this is a prestart job.

The substring retrieval option, which is valid for character data areas only, specifies the starting position and the length of the portion of the data area that is being retrieved into the CL character variable.

Element 3: Starting Position of the Data Area

***ALL:** The entire data area is retrieved.

substring-starting-position: Specify the starting position of the data area being retrieved.

Element 4: Length of the Data Area

substring-length: Specify the length of the data area substring being retrieved.

It is not possible to retrieve data outside the data area. The combination of starting position and length must always specify positions within the data area.

RTNVAR

Specifies the name of the CL program variable that receives the contents of the data area being returned.

No type conversion is performed by the RTVDTAARA command:

- If RTNVAR is declared as TYPE(*DEC), the data area retrieved must be TYPE(*DEC).
- If RTNVAR is declared as TYPE(*CHAR), the data area retrieved must be either TYPE(*CHAR) or TYPE(*LGL).
- If RTNVAR is declared as TYPE(*LGL), the data area retrieved must be either TYPE(*LGL) or TYPE(*CHAR) with a value of either '0' or '1'.

If a retrieved character string is shorter than the length of the variable specified by the RTNVAR parameter, the value is padded on the right with blanks. The retrieved string length must be less than or equal to the CL variable length.

When decimal data areas are retrieved, the decimals are aligned. The value of the integer portion of the data area must fit into the integer portions of the CL variable. Fractional data is truncated if the fraction contains more digits than the CL variable.

Examples for RTVDTAARA

Assume data area DA1 has been created by the following command:

```
CRTDTAARA DTAARA(DA1) TYPE(*CHAR) LEN(3) VALUE(ABC)
```

and variable &CLVAR1 has been declared as:

```
DCL VAR(&CLVAR1) TYPE(*CHAR) LEN(5) VALUE(VWXYZ)
```

Example 1: Running This Command

```
RTVDTAARA DTAARA(DA1) RTNVAR(&CLVAR1)
```

results in:

```
&CLVAR1 = 'ABC '
```

Example 2: Running This Command

```
RTVDTAARA DTAARA(DA1 (2 1)) RTNVAR(&CLVAR1)
```

results in:

```
&CLVAR1 = 'B '
```

Example 3: Decimal Data Area Example

Assume data area DA2 has been created with the following attributes:

```
CRTDTAARA DTAARA(DA2) TYPE(*DEC) LEN(5 2) VALUE(12.39)
```

and variable &CLVAR2 has been declared as:

```
DCL VAR(&CLVAR2) TYPE(*DEC) LEN(5 1) VALUE(4567.8)
```

Example 4: Running This Command

```
RTVDTAARA DTAARA(MYLIB/&DTAARA) RTNVAR(&CLVAR2)
```

results in:

```
CLVAR2 = 0012.3
```

Note that fractional digits are truncated instead of rounded.

Error messages for RTVDTAARA

*ESCAPE Messages

CPF0811

RTNVAR parameter has incorrect length for data area.

CPF0812

RTNVAR parameter type not valid for data area &1.

CPF0813

Value in data area &1 not logical value.

CPF101A

Operation on DDM data area &1 in &2 failed.

CPF1015

Data area &1 in &2 not found.

CPF1016

No authority to data area &1 in &2.

CPF1021

Library &1 not found for data area &2.

CPF1022

No authority to library &1 data area &2.

CPF1043

Boundary alignment for data area not valid.

CPF1044

AREA parameter not valid for data area.

CPF1045

CPYPTR parameter not valid for data area.

- CPF1046**
DTAARA(*GDA) not valid because job not group job.
- CPF1063**
Cannot allocate data area &1 in library &2.
- CPF1067**
Cannot allocate library &1.
- CPF1072**
DTAARA(*PDA) not valid because job not prestart job.
- CPF1087**
Substring not allowed for decimal or logical data area.
- CPF1088**
Starting position outside of data area.
- CPF1089**
Substring specified for data area not valid.
- CPF180B**
Function &1 not allowed.
- » **CPF9899** «
Error occurred during processing of command. «

RTVDSKINF (Retrieve Disk Information) Command Description

RTVDSKINF Command syntax diagram

Purpose

The Retrieve Disk Information (RTVDSKINF) command is used to collect disk space information. The collected information is stored in the library QUSRSYS. » The file name depends on the auxiliary storage pool (ASP) device for which disk space information is retrieved. If the information was retrieved from the system and basic ASPs, the collected information will be stored in file QAEZDISK. If the information was retrieved from an independent ASP device, the collected information will be stored in file QAEZDnnnnn, where 'nnnnn' is the ASP number of the independent ASP. The information will be stored in a data base file member named QCURRENT. «

» Each time this command is run, existing information in QCURRENT is written over. To save existing information in QCURRENT, rename file QAEZDISK or QAEZDnnnnn, or copy the member to another file. «

» To get the most accurate results, this command should be run at a time when there is very little system activity. «

Note:

Do not rename member QCURRENT within QAEZDISK » or QAEZDnnnnn «. If there is more than one member in file QAEZDISK » or QAEZDnnnnn «, the results of running this command can be unpredictable.

» ASPDEV

Specifies the auxiliary storage pool (ASP) device for which disk space information is to be retrieved.

The possible values are:

***SYSBAS:** Disk information for the system ASP and all basic ASPs is retrieved. The system ASP has an ASP number of 1. Basic ASPs have ASP numbers from 2 through 32.

auxiliary-storage-pool-device-name: Specify the name of the ASP device for which disk space information is to be retrieved. Independent ASP devices have ASP numbers greater than 32. You can submit multiple jobs, each performing RTVDSKINF on a different ASP device, to retrieve disk space information for multiple ASP devices in parallel. <<

Example for RTVDSKINF >>

```
RTVDSKINF ASPDEV(*SYSBAS)
```



This command retrieves disk space information of system and basic ASPs, and stores it in member QCURRENT of database file QAEZDISK. Any information in QCURRENT is overwritten.

Error messages for RTVDSKINF

*ESCAPE Messages

CPF1ED1

Not authorized to collect disk space information.

CPF1ED2

File >> &1 << is in use and cannot be accessed.

CPF1E99

Unexpected error occurred.

RTVDOC (Retrieve Document) Command Description

RTVDOC Command syntax diagram

Purpose

The Retrieve Document (RTVDOC) command allows a user to copy information from a specific document to a database file and allows the user to check out a document from the document library. Documents whose storage has been freed (STG(*FREE) specified on the Save Document Library Object (SAVDLO) command) cannot be retrieved.

Restrictions:

1. To retrieve any records from the document to a database file, you must have *USE authority to the document or be working on behalf of a user that has *USE authority to the document.
2. To check out the document, you must have at least *CHANGE authority to the document, or be working on behalf of a user that has *CHANGE authority to the document.
3. To work on behalf of another user, you must have either *ALLOBJ authority or special permission (granted with the Grant User Permission (GRTUSRPMN) command).

Required Parameter

FROMDOC

Specifies the name of the document being retrieved.

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

document-name: Specify the name of the document that is retrieved.

Optional Parameters

FROMFLR

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

***NONE:** No folder name is specified when the document is identified by the DOCID parameter.

folder-name: Specify the name of the folder that contains the document being retrieved.

DOCID

Specifies the library-assigned name of the document being retrieved. This is the name assigned to the document by the system when it is created. The library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command. The library-assigned document name is also returned in a completion message when the File Document (FILDOC) command is used.

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

YYYY = year

MM = month

DD = day

HH = hour

MN = minute


SS = second

HS = hundredths of a second

SNSNSNSN = system name

***NONE:** No library-assigned document name is required when the document is identified by the FROMDOC parameter.

library-assigned-document-name: Specify the library-assigned document name sent. More

information on library-assigned document names is in the Communications Configuration  book.

USRID

Specifies the user ID and address of the user for whom the request is made.

***CURRENT:** The user profile that is currently running is used.

Element 1: User ID

user-ID: Specify the user ID of the user for whom the document is retrieved.

Element 2: User Address

user-address: Specify the user address of the user for whom the document is retrieved.

CHKOUT

Specifies whether the document being retrieved can be replaced with new or changed data. If the document is read only, then specify CHKOUT(*NO). If the document being retrieved cannot be replaced, and CHKOUT(*YES) is specified, an error occurs.

Note:

If a user is retrieving a document to check it out and a cancel request is entered, the document may or may not be checked out.

***NO:** The retrieve request only reads the data. Users requesting this function need only *USE authority to the document. The authority for the public is *READ authority.

***YES:** The document data can be updated and replaced later. Users requesting this function must have *CHANGE authority. The document is unavailable to other users until a replace operation is done using the Replace Document Command (RPLDOC) The checkout flag can be reset with the CHGDOCD or EDTDLOAUT command.

OUTFILE

Specifies the name of the database file where the document data of the display 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 text "OUTFILE created by RTVDOC" is shown and the authority given to users with no specific authority is *EXCLUDE.

Note:

The output file format must be the same as the OSRTVD of the system file QSYS/QAOSIRTV.

More information on defining the format of database files (output files) is in the Office Services Concepts and Programmer's Guide book.

***NONE:** The output is not directed to a database file.

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

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

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

library-name: Specify the name of the library to be searched.

database-file-name: Specify the name of the database file to receive the output.

This file can be reused when other RTVDOC commands are used. Output can be added to the file or can replace the existing records. The IBM-supplied database file QAOSIRTV in library QSYS cannot be specified.

OUTMBR

Specifies the name of the database file member to which the output is directed. If a member already exists, the system uses the second element of this parameter to determine whether the member is cleared before the new records are added. If the member does not exist and a member name is not specified, the system creates a member with the name of the output file specified on the OUTFILE parameter. If an output file member name is specified, but the member does not exist, the system creates it.

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

***REPLACE:** The system clears the existing member and adds the new records.

Note:

Document data for more than one document can be added to the same member. However, an error occurs if the member is filed to another document with the FILDOC command. The FILDOC command only files the first document in the member and issues an error message.

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

OUTDTATYP

Specifies what parts of information about the selected documents are written to the output file if one is specified.

***DFT:** The document information record is written to the output file. The following record codes are written to the output file:

Record Code	Description
105	Document Description
800	Document Data

***ALL:** All information records about the document are written to the output file.

Record Code	Description
105	Document Description
110	Creation Date
115	Expiration date
120	Document date
125	File date
130	Change date
135	Action due date
140	Completion date
145	Author
150	Copy list
155	Document class
160	File cabinet reference
165	Subject
170	Keyword
175	Reference
180	Status
185	Project
500	Interchange document profile data
800	Document Data

***DOCD:** The document information record is written to the output file.

***DOCCLS:** The document class record is written to the output file.

***SUBJECT:** The subject records are written to the output file.

***FILCAB:** The filing cabinet reference code is written to the output file.

***AUTHOR:** The author records are written to the output file.

***KWD:** The keyword records are written to the output file.

***CPYLST:** The copy list records are written to the output file.

***FILDATE:** The file date record is written to the output file.

- ***EXPDATE:** The expiration date record is written to the output file.
- ***DOCDATE:** The document date record is written to the output file.
- ***CRTDATE:** The create date record is written to the output file.
- ***ACTDATE:** The action due date record is written to the output file.
- ***CHGDATE:** The date last changed record is written to the output file.
- ***CMPDATE:** The completion date record is written to the output file.
- ***REF:** The reference record is written to the output file.
- ***STATUS:** The status record is written to the output file.
- ***PROJECT:** The project record is written to the output file.
- ***IDP:** The interchange document profile (IDP) is written to the output file.
- ***DOC:** The document data record is written to the output file.


CMDCHRID

Specifies the character identifier (graphic character set and code page) for data being specified as parameter values on this command. This character identifier (CHRID) is related to the display device used to specify the command. More information about CHRID processing is in the

Application Display Programming  book.

Note:

This value translates the USRID parameter to character set and code page set of '930 256'. The SNA Distribution

Services  book contains the character set and code page table for '930 256'.

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

***DEVD:** The system determines the graphic character set and code page values for the command parameter from the display device description where the command is entered. This option is valid only when specified from an interactive job. If this value is specified in an interactive CL program or a batch job, an error message is sent.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values used to create the command parameter.

Element 2: Code Page

code-page: Specify the code page value used to create the command parameters. Valid values range from 1 through 999.

Examples for RTVDOC

Example 1: Copying All Information

```
RTVDOC FROMDOC(MYDOC) FROMFLR(PERSONAL)
      USRID(*CURRENT) OUTFILE(*CURLIB/MYFILE)
      OUTMBR(*FIRST) MBROPT(*ADD) OUTDTATYP(*ALL)
```

This command copies all information about document MYDOC located in folder PERSONAL for the current user of this command. CHECKOUT(*NO) is assumed; therefore, the document data can only be read. The output is directed to the database file MYFILE in the user's current library and is added to the first member in that file.

Example 2: Copying Default Information

```
RTVDOC FROMDOC(SECOP) FROMFLR(PERSONAL)
      USRID(MARY SYSTEM1 ) CHKOUT(*YES)
      OUTFILE(MARLIB/SECFILE) OUTMBR(*FIRST *ADD)
```

This command copies the default information (*DOCD and *DOC) about document SECOP located in folder PERSONAL for MARY. The document can be updated with new data and then replaced. The current user of this command must have the authority to work on behalf of MARY given by Mary by using the GRTUSRPMN command. The output is directed to the database file SECFILE in Mary's library MARLIB. The output is added to the first member of SECFILE.

Error messages for RTVDOC

*ESCAPE Messages

CPF900B

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

CPF900C

Sign on and verify of user failed.

CPF905C

Error occurred trying to find a translation table.

CPF905F

Retrieval of document from library failed.

CPF9096

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

CPF9860

Error occurred during output file processing.

RTVDLOAUT (Retrieve Document Library Object Authority) Command Description

RTVDLOAUT Command syntax diagram

Purpose

The Retrieve Document Library Object Authority (RTVDLOAUT) command retrieves the authority assigned a folder or filed document.

Restrictions:

1. A user must be in the system distribution directory entry to retrieve the various authorities.
2. A user must have at least *USE authority to the filed document or folder.
3. A user with less than *ALL authority to the filed document or folder will only be able to retrieve that user's authority or the owner.

4. A user must have *ALL or *ALLOBJ authority or be the owner of the filed document or folder to retrieve all the authorities.
5. A user must have *ALLOBJ special authority to retrieve the *ROOT folder public authority.

Required Parameter

DLO Specifies the name of the document or folder for which authorities are retrieved.

***DOCID:** The document or folder is identified using its library-assigned document name specified in the DOCID parameter.

***ROOT:** The root folder contains all first-level folders. If folder *ROOT is specified only public authority will be returned by PUBAUT parameter. No other authorities will be returned.

***LADNTSP:** The timestamp from the library-assigned document name (LADN) specified on the LADNTSP parameter is used to identify the document or folder.

***SYSOBJNAM:** The system object name specified on the SYSOBJNAM parameter is used to identify the document or folder.

document-name: Specify the user-assigned name of the document.

folder-name: Specify the user-assigned name of the folder.

Optional Parameters

FLR Specifies the name of the folder where the object specified on the DLO parameter is located.

***NONE:** The name of folder that contains the object is not specified, the object is not contained in a folder, or the object is specified using the LADNTSP, or SYSOBJNAM parameter, or *ROOT was specified in the DLO parameter.

folder-name: Specify the name of the folder that contains the object.

Note: FLR(*NONE) must be specified if the object is a first-level folder.

DOCID

Specifies the library-assigned name of the document or folder.

***NONE:** The object is not identified using its document identifier (DOCID).

document-identifier: Specify the document identifier of the document or folder. The document identifier is 24 hexadecimal characters in length in the format YYYYMMDDHHMNSSHSSNSNSNSN, where:

YYYY = year

MM = month

DD = day

HH = hour

MN = minute

SS = second

HS = hundredths of a second

SNSNSNSN = system name

LADNTSP

Specifies the LADN timestamp of the document or folder.

***NONE:** The object is not identified using its LADN timestamp.

LADN-timestamp: Specify the LADN timestamp of the document or folder. The LADN timestamp is 16 hexadecimal characters in length in the format YYYYMMDDHHMNSSHS, where:

YYYY = year

MM = month

DD = day

HH = hour

MN = minute

SS = second

HS = hundredths of a second

SYSOBJNAM

Specifies the system object name.

***NONE**: The object is not identified using its system object name.

system-object-name: Specify the 10-character system object name of the document or folder.

STRUSRAUTE

Allows a user to specify the starting user authority entry number to use when retrieving specific user authorities (USRAUT parameter). STRUSRAUTE will enable retrieving specific user authorities for a DLO that has more than 50 specific user authorities. If no CL variable is provided for the USRAUT parameter, this parameter is ignored.

1: User authority will be returned starting with the first specific user authority.

starting-user-authority-entry-number: User authority will be returned starting with the specified entry number. If the value is greater than the number of specific user authorities for the DLO, an error message will be sent and no specific user authorities will be returned.

OWNER

Specifies the name of a 10-character CL variable used to retrieve the owner of the selected document or folder.

AUTL Specifies the name of a 10-character CL variable used to retrieve the authorization list assigned to the selected document or folder. The value ***NONE** is returned if no authorization list has been assigned.

SENSITIV

Specifies the name of a 20-character CL variable used to retrieve the sensitivity assigned to the selected document or folder. The possible values are:

***NONE**: The document has no sensitivity restrictions.

***PERSONAL**: The document is intended for the user as an individual.

***PRIVATE**: The document contains information that should be accessed only by the owner.

***CONFIDENTIAL**: The document contains information that should be handled according to company procedures.

PUBAUT

Specifies the name of a 10-character CL variable used to retrieve the public authority assigned to the selected document or folder. The possible values are:

***USE**: User can view, print, or copy the document or folder.

***CHANGE**: User can perform all operation listed for ***USE** and can also edit and mark the document for offline storage.

***ALL:** User can perform all operations on the document, except change ownership of the document or folder or give themselves authority to work with the document after authority has been revoked.

***EXCLUDE:** All users who are not otherwise authorized to this document or folder are denied access.

USER DEF: Authority to this document is user-defined and is not one of the system-defined sets of authorities (*ALL, *CHANGE, *USE, *EXCLUDE).

***AUTL:** Authority specified in the authorization list being used by this document should determine public authority.

CHKOUTUSR

Specifies the name of a 32-character CL variable used to retrieve the user profile who has the document checked out and the user profile on whose behalf the document was checked out. If no user has checked out the document, the 32-character variable will be blank. If the document was not checked out by a user working on behalf of another user, the last 16 characters will be blanks.

Character position	Length	Description
1	8	Checked out by user ID
9	8	Checked out by user address
17	8	Checked out for user ID
25	8	Checked out for user address

ACC Specifies the name of a 220-character CL variable used to retrieve the access codes assigned to the specified document or folder.

Character position	Length	Description
1	5	Total number of access codes
6	15	Reserved
21	4	1st access code
25	4	2nd access code
29	4	3rd access code
33 ... 220	4	nth access code

USRAUT

Specifies the name of a 1020-character CL variable used to retrieve the specific user authority assigned to the document or folder. USRAUT will return a maximum of 50 specific authorities per invocation of the RTVDLOAUT command. If the document or folder has more than 50 specific user authorities associated with it, you can use the STRUSRAUTE parameter on subsequent invocations of RTVDLOAUT to return user authorities starting with the specified entry numbers.

Within the 1020-character CL variable are the total number of authorized users, the number of authority entries returned, the starting entry number and the ending entry number.

```
RTVDLOAUT DLO(MYDOC) FLR(MYFLR)
USRAUT(&RTNUSRAUT) + STRUSRAUTE(1)
```

In the example above, MYDOC has 55 authorized users. The total number of specific user authorities will be 55. The number of authority entries returned will be 50. The starting authority entry number will be 1. The ending authority entry number will be 51.

Character position	Length	Description
1	5	Total number of specific user authorities

Character position	Length	Description
6	5	Number of authority entries returned
11	5	Starting authority entry number
16	5	Ending authority entry number
21	10	1st user profile name
31	10	1st specific authority
41	10	2nd user profile name
51	10	2nd specific authority
61 ... 1020

GRPAUT

Specifies the name of a 340-character CL variable used to retrieve the group authority assigned to the specified document or folder. Only the groups associated to the user of this command will be returned.

Character position	Length	Description
1	5	Total number of group authorities
6	15	Reserved
21	10	1st group user profile name
31	10	1st group authority
41	10	2nd group user profile name
51	10	2nd group authority
61 ... 340

PGP Specifies the name of a 10-character CL variable used to retrieve the primary group assigned to the specified document or folder. The value *NONE is returned if no primary group has been assigned.

Example for RTVDLOAUT

```
RTVDLOAUT DLO(MYDOC) FLR(MYFLR) OWNER(&OWNER)
```

This command finds the document library object MYDOC in folder MYFLR and returns its owner in the variable &OWNER.

```
RTVDLOAUT DLO(MYDOC) FLR(MYFLR) USRAUT(&RTNUSRAUT) + STRUSRAUTE(1)
```

Where document MYDOC in MYFLR has 55 authorized users.

Returned in the &RTNUSRAUT variable, the total number of specific authorities will be 55. The number of authority entries returned will be 50. The starting authority entry number will be 1. The ending authority entry number will be 51. Following will be the list of 50 privately authorized users and their authorities.

No error messages.

RTVDLONAM (Retrieve Document Library Object Name) Command Description

RTVDLONAM Command syntax diagram

Purpose

The Retrieve Document Library Object Name (RTVDLONAM) command is used in a CL program or REXX procedure to retrieve a list of all forms of a name for a folder, filed document, or distribution document.

Restrictions:

1. A user must have *USE authority to the filed document or folder to retrieve the various forms of the name.
2. A user must have *ALLOBJ authority to retrieve the various forms of the name for a distribution document.

Required Parameter

DLO Specifies the name of the document or folder for which forms of the name are retrieved.

***DOCID:** The library-assigned name specified on the DOCID parameter is used to identify the document or folder.

***LADNTSP:** The timestamp from the library-assigned document name (LADN) specified on the LADNTSP parameter is used to identify the document or folder.

***SYSOBJNAM:** The system object name specified on the SYSOBJNAM parameter is used to identify the document or folder.

document-name: Specify the user-assigned name of the document.

folder-name: Specify the user-assigned name of the folder.

Optional Parameters

FLR Specifies the name of the folder where the object specified on the DLO parameter is located.

***NONE:** The name of folder that contains the object is not specified, the object is not contained in a folder, or the object is specified using the DOCID, LADNTSP, or SYSOBJNAM parameter.

folder-name: Specify the name of the folder that contains the object.

Note: FLR(*NONE) must be specified if the object is a first-level folder.

DOCID

Specifies the library-assigned name of the document or folder.

***NONE:** The object is not identified using its library-assigned name.

document-ID: Specify the library-assigned name of the document or folder object. Library-assigned names are 16 to 24 characters in length in the format YYYYMMDDHHMNSSHSSNSNSNSN, where:

YYYY = year

MM = month

DD = day

HH = hour

MN = minute

SS = second

HS = hundredths of a second

SNSNSNSN = system name (from 1 through 8 characters in length or omitted)

LADNTSP

Specifies the LADN timestamp of the document or folder.

***NONE:** The object is not identified using its LADN timestamp.

LADN-timestamp: Specify the LADN timestamp of the document or folder. The LADN timestamp is 16 hexadecimal characters in length in the format YYYYMMDDHHMNSSHS, where:

YYYY = year

MM = month

DD = day

HH = hour

MN = minute

SS = second

HS = hundredths of a second

SYSOBJNAM

Specifies the system object name.

***NONE**: The object is not identified using its system object name.

system-object-name: Specify the 10-character system object name of the document or folder.

OBJCLS

Specifies the class of the object to locate.

***DOC**: The object is a filed document.

***FLR**: The object is a folder.

***DST**: The object is a distribution document.

RTNDLO

Specifies the name of a 12-character CL variable used to retrieve the user-assigned name of the selected document or folder. The *NONE value is returned for a distribution document or a document without a folder.

RTNFLR

Specifies the name of a 63-character CL variable used to retrieve the folder path of the selected document or folder. The *NONE value is returned for a distribution document, a document without a folder, or a first-level folder.

RTNDOCID

Specifies the name of a 24-character CL variable used to retrieve the library-assigned document name of the selected object. The variable is in the format YYYYMMDDHHMNSSHSSNSNSNSN. The *NONE value is returned for a distribution document.

RTNLADNTSP

Specifies the name of a 16-character CL variable used to retrieve the timestamp from the LADN of the selected object. The variable is in the format YYYYMMDDHHMNSSHS.

RTNOBJNAM

Specifies the name of a 10-character CL variable used to retrieve the system object name of the selected object.

RTNOBJCLS

Specifies the name of a 8-character CL variable used to retrieve the object class. A value of *DOC is returned for a filed document, *FLR for a folder, and *DST for a distribution document.

RTNASP

Specifies the name of a variable used to return the auxiliary storage pool ID. In control language (CL) programs, this should be a decimal variable of length (2 0). The following values can be returned:

1 The object is in the system auxiliary storage pool.

2-16 The object is in a user auxiliary storage pool.

RTNOVRFLW

Specifies the name of a 1-character CL variable used to retrieve the overflow status of the object, where:

N = No, the object has not overflowed its ASP

Y = Yes, the object has overflowed its ASP and part or all of the object resides in the system ASP

Example for RTVDLONAM

```
RTVDLONAM DLO(MYDOC) FLR(MYFLR) OBJCLS(*DOC)
RTNDOCID(&DOCID)
```

This command finds the document MYDOC in folder MYFLR and returns its document identifier in the variable &DOCID.

Error messages for RTVDLONAM

***ESCAPE Messages**

CPF8AC0

&1 command failed.

CPF8AC1

Not authorized to distribution documents.

CPF8A75

Not authorized to access folder &1.

CPF8A77

Folder &1 not found.

CPF8A82

Document &2 not found in folder &1.

CPF8A83

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

RTVGRPA (Retrieve Group Attributes) Command Description

RTVGRPA Command syntax diagram

Purpose

The Retrieve Group Attributes (RTVGRPA) command is used in a CL program or REXX procedure to retrieve information about the group in which the job that issued the RTVGRPA command belongs. The following attributes can be retrieved:

- The group job name of the job calling the RTVGRPA command
- A list that contains information about all active jobs in the group
- A count of the number of active jobs in the group
- The name of the group message queue
- The library where the group message queue resides
- The group job name and job number of the previously active job in the group
- A control code indicating why the currently active job in the group gained control

The prompt for this command lists the minimum length for retrieved variables next to the appropriate parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length, and the second number indicates the minimum number of decimal positions.

Optional Parameters

GRPJOB

Specifies the name of the CL variable that receives the group job name of the job. The variable must be a character variable with a minimum length of 10 characters. If the group job name has fewer characters than the variable allows, the value is padded on the right with blanks.

Note:

This is the only parameter of the RTVGRPA command that is meaningful if the job that issued the RTVGRPA command is not a group job. When this occurs, the variable is set to the special value of *NONE, and other parameters specified in the command are filled with zeros or blanks (numeric variables with zeros, character variables with blanks). If this parameter is not specified and the job issuing this command is not a group job, an error message is sent.

GRPJOB

Specifies the name of the CL variable that receives the list of jobs in the group. Each entry contains the job's group job name, the job number, and the 50 characters of descriptive text.

The maximum number of entries in the variable is 16. The entries are ordered by most recently active job (the group job that is issuing the RTVGRPA command is always first in the variable).

The variable must be a character variable with a minimum length of 1056 characters. If the group job list has fewer characters than the variable allows, the value is padded on the right with blanks.

The group job list has the following format:

Entry-1

Group-job-name
CHAR(10)

Job-number
CHAR(6)

Group-job-text
CHAR(50)

Entry-2

.
. .
.

Entry-16

Group-job-name
CHAR(10)

Job-number
CHAR(6)

Group-job-text
CHAR(50)

This list contains all of the active jobs in the group. Jobs in the group that have not completely ended (jobs that have been canceled) are not listed.

GRPJOBcnt

Specifies the CL variable that receives the count of active jobs in the group. The CL variable must be a three-position decimal variable with no decimal positions.

The CL variable contains the number of nonblank entries in the group job list. The count includes all of the active jobs in the group. Jobs that have not completely ended (jobs that have been canceled) are not counted.

MSGQ

Specifies the name of the CL variable that receives the group message queue name. This must be a character variable with a minimum length of 10 characters. If the message queue name has fewer characters than the variable allows, the value is padded on the right with blanks.

MSGQLIB

Specifies the name of the CL variable that receives the name of the library that contains the group message queue. This variable must be a character variable with a minimum length of 10 characters. If the library name has fewer characters than the variable allows, the value is padded on the right with blanks. If there is no message queue associated with the group, the CL variable is set to blanks.

Refer to the Change Group Attributes (CHGGRPA) command description for more information on the use of the group message queue.

PRVGRPJOB

Specifies the name of the CL variable that receives the group job name and job number of the previously active job in the group. The variable must be a character variable with a minimum length of 16 characters. If the group job name has fewer characters than the variable allows, the value is padded on the right with blanks.

If there is no previously active job in the group (no Transfer to Group Job [TFRGRPJOB] command is issued), the group job name portion of the CL variable is set to the special value of *NONE, and the job number portion of the CL variable is set to blanks. The CL variable is returned in the following format:

Group-job-name	CHAR(10)
Job-number	CHAR(6)

Even though a group job name and job number are returned, it is not necessarily true that the group job still exists. There are cases in which the previously active job in the group has ended, which caused the currently active job to become active.

CTLCD

Specifies the name of the CL variable that receives information about why the active job in the group has gained control. The CL variable must be a three-position decimal variable with no decimal positions. The following control codes (and their meanings) are possible:

- 0** There was no previously active job (no TFRGRPJOB commands have been run for this group).
- 10** The previously active job selected this job to be transferred to on the TFRGRPJOB command.
- 20** The previously active job's first group program ended normally, and this job was the most recently active job in the group.

- 30 The previously active job was ended by the End Group Job (ENDGRPJOB) command, and this job was selected to gain control (the RSMGRPJOB parameter specified this group job).
- 40 The previously active job was ended by the End Group Job (ENDGRPJOB) command and selected a job other than this job to gain control (which was ended before it could be resumed). Since this job was the most recently active job in the group, control is passed to it.
- 50 The previously active job was ended by the End Group Job (ENDGRPJOB) command, and this job was the most recently active job in the group (the RSMGRPJOB parameter specified *PRV).
- 60 The previously active job's first group program ended abnormally, and this job was the most recently active job in the group.
- 70 The previously active job was ended by the End Job (ENDJOB) command, and this job was the most recently active job in the group.

Examples for RTVGRPA

Assume jobs 0001/QUSER/WORKST01 and 0002/QUSER/WORKST01 are group jobs with group job names GROUPJ1 and GROUPJ2, respectively. Also assume that message queue QGPL/GROUPMSGQ is associated with the group. If group job GROUPJ1 has just issued the TFRGRPJOB command to transfer to group job GROUPJ2, and GROUPJ2 called the following CL program:

PGM Example

```
DCL VAR(&GRPJOB) TYPE(*CHAR) LEN(10)
DCL VAR(&GRPJOB) TYPE(*CHAR) LEN(1056)
DCL VAR(&GRPCOUNT) TYPE(*DEC) LEN(3 0)
DCL VAR(&MSGQNAME) TYPE(*CHAR) LEN(10)
DCL VAR(&MSGQLIB) TYPE(*CHAR) LEN(10)
DCL VAR(&PRVJOB) TYPE(*CHAR) LEN(16)
DCL VAR(&CTLCODE) TYPE(*DEC) LEN(3 0)
```

```
RTVGRPA GRPJOB(&GRPJOB) GRPJOB(&GRPJOB)
GRPJOBcnt(&GRPCOUNT) MSGQ(&MSGQNAME)
MSGQLIB(&MSGQLIB) PRVGRPJOB(&PRVJOB)
CTLCODE(&CTLCODE)
```

The contents of the CL variables returned are as follows:

```
&GRPJOB: GROUPJ2
&GRPJOB:
```

```
GROUPJ2 0002 50 characters of text for
           this group job...
GROUPJ1 0001 50 characters of text for
           this group job...
```

Fourteen more entries, full of blanks

```
&GRPCOUNT: 002
&MSGQNAME: GROUPMSGQ
&MSGQLIB: QGPL
&PRVJOB: GROUPJ1 0001
&CTLCODE: 010
```

Error messages for RTVGRPA

*ESCAPE Messages

CPF1309

Subsystem cannot complete the &1 command.

CPF1311

Job is not a group job.

CPF1317

No response from subsystem for job &3/&2/&1.

CPF1351

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

RTVJOBA (Retrieve Job Attributes) Command Description

RTVJOBA Command syntax diagram

Purpose

The Retrieve Job Attributes (RTVJOBA) command is used in a CL program or REXX procedure to retrieve the values of one or more job attributes and to place those values into the specified CL variable. The attributes are retrieved for the job in which this command is used. If a job attribute does not exist because the attribute is now supported at the thread level only (e.g. CURUSER), then the thread attribute is retrieved for the thread in which this command is used.

The CL prompt for this command lists the minimum length for retrieved variables next to the appropriate parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

1. This command is conditionally threadsafe. Refer to "Usage Notes" on page 154 for additional information.

Optional Parameters

JOB Specifies, if the job name is returned, the name of the CL variable that receives the name of the job. The variable must be a character variable with a minimum length of 10 characters. If the job name has fewer characters than the variable allows, the value is padded on the right with blanks.

USER Specifies, if the user name is returned, the name of the CL variable that receives the name of the user profile associated with the job when the job was started. The user name is the second part of the qualified job name. The variable must be a character variable with a minimum length of 10 characters. If the user name has fewer characters than the variable allows, the value is padded on the right with blanks.

NBR Specifies, if the job number is being returned, the name of the CL variable that receives the unique 6-character number assigned to the job by the system. The job number is the first part of the qualified job name.

LOGLVL

Specifies the name of the CL variable that receives the 1-character value, ranging from 0 through 4, that is the message logging level used to determine the type of message information logged in the job log. The variable must be a character variable with a minimum length of 1 character.

LOGSEV

Specifies the name of the CL variable that receives the 2-digit value, ranging from 00 through 99, which is the severity level that is used in conjunction with the logging level to determine which error messages are logged in the job log. The variable must be a 2-digit decimal variable specified with no decimal positions.

LOGTYPE

Specifies the name of the CL variable that receives the special value that indicates the level of text that appears for any message written to the job log. The variable must be a character variable with a minimum length of 10 characters. If the special value has fewer characters than the variable allows, the value is padded on the right with blanks.

LOGCLPGM

Specifies the name of the CL variable that receives the special value that indicates whether processed commands in a CL program are being logged in the job log. Note that this logging flag only has meaning for commands processed in CL programs which were created with LOG(*JOB) specified on the Create CL Program (CRTCLPGM) command. Additional information on the job's CL program logging flag is in the LOGCLPGM parameter description in the Change Job (CHGJOB) command. The variable must be a character variable with a minimum length of 10 characters. If the special value has fewer characters than the variable allows, the value is padded on the right with blanks.

INQMSGRPY

Specifies the name of the CL variable that receives the special value that indicates how inquiry messages are being handled by the job. Additional information on the value returned in the variable is in the INQMSGRPY parameter description in the Change Job (CHGJOB) command. The variable must be a character variable with a minimum length of 10 characters. If the special value has fewer characters than the variable allows, the value is padded on the right with blanks.

OUTQ Specifies the name of the CL variable that receives the name of the output queue being used by the job for spooled output. The variable must be a character variable with a minimum length of 10 characters. If the output queue name has fewer characters than the variable allows, the value is padded on the right with blanks.

The special value *DEV can be retrieved. This value indicates that the output queue with the same name as the printer device, which is specified on either the printer file or the job description, will be used for the spooled output of this job.

OUTQLIB

Specifies the name of the CL variable that receives the name of the library that contains the output queue being used by the job for spooled output. The library name is the second part of the qualified output queue name (*output-queue/library-name*). The variable must be a character variable with a minimum length of 10 characters. If the output queue library name has fewer characters than the variable allows, the value is padded on the right with blanks.

ACGCDE

Specifies the name of the CL variable that receives the accounting code for the job. The variable must be a character variable with a minimum length of 15 characters. If the accounting code has fewer characters than the variable allows, the value is padded on the right with blanks.

DATE Specifies, if the job date is to be returned, the name of the CL variable that receives the date assigned to the job by the system when the job was started. The variable must be a character variable with a minimum length of 6 characters. The job date is returned in the date format for the job.

SWS Specifies, if the job switch settings are being returned, the name of the CL variable that receives the status (on or off) value of the eight job switches used by the job. The job switches are retrieved as a single 8-character value with each of the characters specifying a 1 (on) or 0 (off) as the value of the associated switch. The CL variable must be a character variable with a minimum length of 8 characters.

TYPE Specifies, if the type of job environment is being returned, the name of the CL variable that receives the 1-character value representing the environment of the job. A character value of 0 indicates the job is running as a batch job, and a 1 indicates an interactive job. The variable must be a character variable with a minimum length of 1 character.

RTNCDE

Specifies (if the completion status of an RPG, COBOL, DFU, or sort utility program is being returned) the name of the CL variable that receives the 5-digit decimal return code. The return code is set by these programs before they return to the programs that called them. The return code indicates the completion status of the last program (of these types) that has completed processing within the job, as follows:

- 0** Normal return (RPG, COBOL, DFU, or sort Utility)
- 1** LR (last record) indicator on (RPG)
- 2** Error-no halt indicator set (RPG, COBOL, DFU, or sort Utility)
- 3** Halt indicator set on (one of the RPG indicators H1 through H9)

The CL variable must be a five-position decimal variable with no decimal positions.

ENDSTS

Specifies, if checking for a controlled end operation, the name of the CL variable that receives the end status. The single-character value indicates whether a controlled end that affects the job is currently being performed. A value of 1 indicates that the system, the subsystem in which the job is running, or the job itself is being ended; a value of 0 indicates no controlled end is being performed. The CL variable must be a character variable with a minimum length of 1 character.

RUNPTY

Specifies the name of the CL variable that receives the 2-digit value, ranging from 1 through 99, that is the run (or processing) priority for the job in which this command is used. This value is the highest run priority allowed for any thread within the job. Individual threads within the job may have a lower run priority. To retrieve the run priority for the thread in which one is running, use the Retrieve Current Attributes (QWTRTVCA) API. For additional information on run priority, refer to this parameter description in the Change Job (CHGJOB) command. The variable must be a 2-digit decimal variable specified with no decimal positions.

TIMESLICE

Specifies the name of the CL variable that receives the 7-digit value, ranging from 8 through 9999999, that is the maximum number of milliseconds that a thread within this job can run when it is given processing time. For additional information on time slice, refer to this parameter description under the Change Job (CHGJOB) command. The variable must be a 7-digit decimal variable specified with no decimal positions.

PURGE

Specifies the name of the CL variable that receives the special value that indicates whether the job is marked as eligible to move from main storage at the end of a time slice or a long wait state. For additional information on job purging, refer to this parameter description under the Change Job (CHGJOB) command. The variable must be a character variable with a minimum length of 10 characters. If the special value has fewer characters than the variable allows, the value is padded on the right with blanks.

DFTWAIT

Specifies the name of the CL variable that receives the 7-digit value, ranging from 1 through 9999999 (or -1 if the value is set to *NOMAX), which is the default for the maximum number of seconds that the system waits for a machine instruction to acquire a resource. For additional information on wait-time, refer to the DFTWAIT parameter description under the Change Job (CHGJOB) command. The variable must be a 7-digit decimal variable specified with no decimal positions.

USRLIBL

Specifies the name of the CL variable that receives the user portion of the thread's library list. The

variable must be a character variable with a minimum of 275 characters. The variable must be large enough to hold all of the libraries found in the user portion of the library list or an error occurs. If there are more than 25 libraries, the variable should be declared with a minimum length of 2750 characters. Each library name returned is left-justified in an 11-character field and padded on the right with blanks.

SBMMSGQ

Specifies the name of the CL variable that receives the name of a message queue. If this parameter is coded for a RTVJOBA command in a CL program which is part of a batch job, the value returned is associated with the MSGQ parameter on the JOB or SBMJOB command which caused the batch job to start running. If *NONE was specified for the MSGQ parameter of the JOB or SBMJOB commands, or the CL program that contains the RTVJOBA command is not part of a batch job, *NONE is returned to the CL variable coded for this parameter. The variable must be a character variable with a minimum length of 10 characters. If the message queue name has fewer characters than the variable allows, the value is padded on the right with blanks.

SBMMSGQLIB

Specifies the name of the CL variable that receives the name of the library that contains the message queue described above. (See the SBMMSGQ parameter description above.) The library name is the second part of the qualified message queue name. If *NONE is returned for the SBMMSGQ parameter, it is also returned for this parameter. The variable must be a character variable with a minimum length of 10 characters. If the message queue library name has fewer characters than the variable allows, the value is padded on the right with blanks.

PRTTXT

Specifies the name of the CL variable that receives the print text for the job. This must be a character variable with a minimum length of 30 characters. More information on this parameter is in Commonly used parameters.

DDMCNV

Specifies the name of a CL variable that receives the special value that indicates the action taken for DDM conversations on the job. The variable must be a character variable with a minimum length of 5 characters. If the special value has fewer characters than the variable allows, the value is padded on the right with blanks.

BRKMSG

Specifies the name of a CL variable that receives the special value that indicates the mode for break message handling that is in effect for the job. The variable must be a character variable with a minimum length of 7 characters. If the special value has fewer characters than the variable allows, the value is padded on the right with blanks.

DATFMT

Specifies the name of a CL variable that receives the special value being used as the date format for the job. The variable must be a character variable with a minimum length of 4 characters. If the special value has fewer than 4 characters, the value is padded on the right with blanks.

DATSEP

Specifies the name of a CL variable that receives the character being used as the date separator character for the job. The variable must be a character variable with a minimum length of 1 character.

CURLIB

Specifies the name of a CL variable that is used to retrieve the name of the current library for the thread. The variable must be a character variable with a minimum length of 10 characters. If the current library name has fewer characters than the variable allows, the value is padded on the right with blanks.

Note:

If the thread does not have a current library, a value of *NONE is returned in this variable.

PRTDEV

Specifies the name of the CL variable that receives the name of the printer device. The variable must be a character variable with a minimum length of 10 characters. If the printer device name has fewer characters than the variable allows, the value is padded on the right with blanks.

SYSLIBL

Specifies the name of the CL variable that receives the system portion of the thread's library list. The variable must be a character variable with up to 165 characters. Each library name returned is left-justified in an 11-character field and padded on the right with blanks.

CURUSER

Specifies, if the user name is returned, the 10-character CL variable that receives the name of the current user profile. If the current user name has fewer characters than the variable allows, the value is padded on the right with blanks.

SUBTYPE

Specifies, if the SUBTYPE of job environment is being returned, the 1-character CL variable that receives the name of the job environment. The valid character values are:

Character**Value**

*	Job has no subtype
E	Job is running as an evoked job
T	Job is running as a Multiple Requester Terminal (MRT) job
J	Job is running as a prestart job
P	Job is running as a print driver

PRTKEYFMT

Specifies the name of the CL variable that receives the print key format for the job. The variable must have a minimum length of 7 characters. The special value *NONE, *PRTBDR, *PRTHDR, or *PRTALL is returned.

TIMSEP

Specifies the name of a CL variable that receives the character being used as the time separator character for the job. The variable must be a character variable with a minimum length of one character.

TSEPOOL

Specifies the name of the CL variable that receives the special value indicating whether interactive jobs are moved to another main storage pool when they reach the time slice end. The variable must be a character variable with a minimum length of 10 characters.

DEVRCYACN

Specifies the name of the CL variable that receives the special value indicating the recovery action to take for the job when an I/O error is encountered on the *REQUESTER device for interactive jobs. The variable must be a character variable with a minimum length of 13 characters.

STSMMSG

Specifies the name of the CL variable that receives the special value indicating how status messages are handled for the job. The variable must be a character variable with a minimum length of 7 characters.

SRTSEQ

Specifies the name of the CL variable that receives the name of the sort sequence table used for the job. The special value *LANGIDUNQ, *LANGIDSHR, or *HEX can be returned to the variable. The variable must be a character variable with a minimum length of 10 characters.

SRTSEQLIB

Specifies the name of the CL variable that receives the name of the library containing the sort sequence table to be used for the job. The variable must be a character variable with a minimum length of 10 characters. If SRTSEQ is *LANGIDUNQ, *LANGIDSHR, or *HEX, blanks are returned in the variable.

LANGID

Specifies the name of the CL variable that receives the value indicating the language identifier to be used for the job. The variable must be a character variable with a minimum length of 3 characters.

CNTRYID

Specifies the name of the CL variable that receives the value indicating the country or region identifier to be used for the job. The variable must be a character variable with a minimum length of 2 characters.

CCSID

Specifies the name of a CL variable that receives the CCSID value being used for the job. The variable must be a 5-digit decimal variable with no decimal positions.

JOBMSGQMX

Specifies the name of a CL variable that receives the maximum size of the job message queue. The variable must be a 2-digit decimal variable with no decimal position.

JOBMSGQFL

Specifies the action that should be taken when the job message queue is full. The variable must have a minimum length of 10 characters. The special value *NOWRAP, *WRAP, or *PRTWRAP is returned.

DFTCCSID

Specifies the name of a CL variable that receives the DFTCCSID value being used for the job. The variable must be a 5-digit decimal variable with no decimal positions.

CYMDDATE

Specifies the name of the CL variable that receives the date assigned to the job by the system when the job was started. The variable must be a character variable with a minimum length of 7 characters. The job date is returned in the format CYYMMDD, where C is the century, YY is the year, MM is the month and DD is the day.

DECfmt

Specifies the name of a CL variable that receives the character being used as the decimal format for the job. The variable must be a character variable with a minimum length of 1 character.

CHRIDCTL

Specifies the name of a CL variable that receives the value being used as the character identifier control for the job. The variable must be a character variable with a minimum length of 10 characters.

▶ ASPGRP

Specifies the name of a CL variable that receives the auxiliary storage pool (ASP) group name. The variable must be a character variable with a minimum length of 10 characters. The special value *NONE is returned if there is no ASP group for the thread.

SPLFACN

Specifies the name of a CL variable that receives the special value indicating whether spooled files are kept with the job or detached when the job ends. The variable must be a character variable with a minimum length of 10 characters. The special value *KEEP or *DETACH is returned. <<

Example for RTVJOBA

```
RTVJOBA NBR(&JOBNBR) DATE(&JOBDATE) DFTCCSID(&DFTCCSID)
```

This command retrieves the job number and job date attributes from the job in which this command is located. The 6-digit job number is copied into the CL variable &JOBNBR. The job date is copied into the CL variable &JOBDATE; the values for both &JOBNBR and &JOBDATE must be 6 characters in length. The 5-digit DFTCCSID value is copied into the CL variable &DFTCCSID; this value must be 5 characters in length. The format of the date is determined by the contents of the system value QDATFMT, which controls the system date format.

```
/* Declare Variables */
DCL &LIBL *CHAR 2750
DCL &CHGLIBL *CHAR 2760

/* save library list */
RTVJOBA USRLIBL(&LIBL)
*
*
/* temporarily change library list */
CHGLIBL LIBL(MYLIB QGPL)
*
*
/* build command string */
CHGVAR &CHGLIBL ('CHGLIBL (' *CAT &LIBL *TCAT ')')

/* restore library list */
CALL QCMDEXC (&CHGLIBL 2760)
```

The above command retrieves the user portion of the library list so that it later can be restored from its temporary state, where only MYLIB and QGPL were in the user portion of the library list, to its original state.

If there are no libraries on the user portion of the library list, blanks are returned in the variable. If a library on the library list has been deleted, the value '*DELETED' is put in the variable position for that name.

Usage Notes

Considerations for Attributes Scope and Thread Safety

This command is intended to be used to retrieve job attributes for the job in which the command is used. However, if a job attribute does not exist because the attribute is now supported at the thread level only, then the thread attribute is retrieved for the thread in which this command is used.

The Scope column of Table 1 (page 155) shows whether the attribute is scoped to the **Job** to the **Thread**.

The Threadsafe column of Table 1 (page 155) indicates if the attributes are considered to be threadsafe.

Yes Attributes marked with a **Yes** can be retrieved safely from either the initial thread or from a secondary thread.

No An attribute marked with this value is not threadsafe, and should not be retrieved in a multi-threaded job.

Table 1. Attribute, Scope, and Thread Safety

Attribute	Scope	Threadsafe
▶ ASP group name (ASPGRP)	Thread	Yes ◀
Break message handling (BRKMSG)	Job	Yes
Character identifier control (CHRIDCTL)	Job	Yes
Coded character set ID (CCSID)	Job	Yes
Country or region ID (CNTRYID)	Job	Yes
Current library (CURLIB)	Thread	Yes
Current user profile (CURUSER)	Thread	Yes
Date format (DATFMT)	Job	Yes
Date separator (DATSEP)	Job	Yes
DDM conversion handling (DDMCNV)	Job	Yes
Decimal format (DECfmt)	Job	Yes
Default coded character set identifier (DFTCCSID)	Job	Yes
Default wait (DFTWAIT)	Job	Yes
Device recovery action (DEVRCYACN)	Job	Yes
End status (ENDSTS)	Job	Yes
Inquiry message reply (INQMSGRPY)	Job	Yes
Job accounting code (ACGCDE)	Job	Yes
Job date with century (CYMDDATE)	Job	Yes
Job date without century (DATE)	Job	Yes
Job message queue full action (JOBMSGQFL)	Job	Yes
Job message queue maximum size (JOBMSGQMX)	Job	Yes
Job name (JOB)	Job	Yes
Job number (NBR)	Job	Yes
Job subtype (SUBTYPE)	Job	Yes
Job switches (SWS)	Job	Yes
Job type (TYPE)	Job	Yes
Language ID (LANGID)	Job	Yes
Logging Level (LOGLVL)	Job	Yes
Logging of CL programs (LOGCLPGM)	Job	Yes
Logging Severity (LOGSEV)	Job	Yes
Logging Text (LOGTYPE)	Job	Yes
Output queue library name (OUTQLIB)	Job	Yes
Output queue name (OUTQ)	Job	Yes
Print key format (PRTKEYFMT)	Job	Yes
Print text (PRTTXT)	Job	Yes
Printer device name (PRTDEV)	Job	Yes
Program return code (RTNCDE)	Job	No
Purge (PURGE)	Job	Yes
Run priority (RUNPTY)	Job	Yes
Sort sequence (SRTSEQ)	Job	Yes
Sort sequence library (SRTSEQLIB)	Job	Yes
▶ Spooled file action (SPLFACN)	Job	Yes ◀
Status message handling (STSMSG)	Job	Yes
Submitter's message queue library name (SBMMSGQLIB)	Job	Yes
Submitter's message queue name (SBMMSGQ)	Job	Yes
System library list (SYSLIBL)	Thread	Yes
Time separator (TIMSEP)	Job	Yes
Time slice (TIMESLICE)	Job	Yes
Time-slice end pool (TSEPOOL)	Job	Yes
User library list (USRLIBL)	Thread	Yes
User name (USER)	Job	Yes

Error messages for RTVJOB

*ESCAPE Messages

CPF098A

USRLIBL parameter size is too small.

CPF9899

Error occurred during processing of command.

RTVJRNE (Retrieve Journal Entry) Command Description

RTVJRNE Command syntax diagram



Purpose

The Retrieve Journal Entry (RTVJRNE) command is used in a CL program or REXX procedure to retrieve a journal entry and place the results in CL variables. The CL variables contain information, such as the sequence number of the retrieved entry, and are useful in automating certain types of recovery functions. The search for a journal entry can be restricted to an object, to a range of journal receivers, to a range of journal entries, to a journal code, to an entry type, to a job, to a program, to a user profile, or to a commit cycle identifier. Multiple limitation criteria can be specified. If more than one journal entry satisfies the search values specified, the first occurrence of a journal entry satisfying all of the specified search values is returned. If there is no journal entry satisfying the search values specified, the command ends with an escape message, and the return CL variables (RTNSEQNBR, RTNJRNCD, RTNENTTYP, RTNRCV, RTNRCVLIB and RTNJRNE) remain the same.

The order of the search through the journal entries can be ascending or descending. The search order is determined by the value specified in the SEARCH parameter. The value for the FROM parameter must come before the value specified for the TO parameter in the specified search order.

The CL prompt for this command lists the minimum length for retrieved variables next to the correct parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length, and the second number indicates the minimum number of decimal positions.

Restrictions:

1. If the sequence number is reset in the range of receivers specified, the first occurrence of either the FROMENT or TOENT parameter is used, if they are specified.
2. The FILE, JRNCD, ENTYP, JOB, PGM, USRPRF, CMTCYCID, and DEPEND parameters can be used to specify a subset of all available entries within a range of journal entries.
 - If no values are specified using these parameters, all available journal entries are retrieved.
 - If more than one of these parameters are specified, then a journal entry must satisfy all of the values specified on these parameters, except when *IGNFILSLT is specified on the JRNCD parameter.
 - If a journal code is specified on the JRNCD parameter and *IGNFILSLT is specified for 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.
3. 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 a journal that had  a RCVSIZOPT or FIXLENTA option specified that omitted the collection of that data. 
4. If more than the maximum number of objects is identified (32,767 objects), an error occurs and no entries are retrieved. This restriction is ignored if *ALLFILE is specified.

Required Parameter

JRN Specifies the qualified name of the journal from which the journal entry is retrieved.

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

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

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

library-name: Specify the name of the library to be searched.

journal-name: Specify the name of the journal from which the journal entry is retrieved.

Optional Parameters

FILE Specifies a maximum of 300 qualified file names whose journal entries are retrieved. This parameter also specifies the name of the file member whose journal entries are to be retrieved.

To determine which journal entries are to be retrieved, based on the specified file member name, the following is done:

- If the journal is a local journal, and if the specified file member currently exists on the system, the journal identifier is determined from the specified file member. All journal entries in the specified receiver range for that journal identifier are retrieved.
- If the journal is a remote journal, or if the specified file member does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified file member. All journal entries in the specified receiver range for those journal identifiers are retrieved.

There can be more than one journal identifier associated with the specified file member if, for example, a file member was created by that name, it was journaled, and then deleted. Then another file member was created with the same name, and it was also journaled and then deleted. All of these actions would have to occur within the specified receiver range.

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 article in the Information Center for more information.
2. When specifying a database file on this parameter, journal entries with the following journal code values are retrieved only if they satisfy the values specified on the other parameters:
 - Journal code D (database file-level information entries).
 - Journal code F (file member-level information entries).
 - Journal code R (record-level information entries).
 - Journal code U (user-generated entries).
 - Other journal codes, if *IGNFILSLT is specified on that journal code. If *ALLSLT is specified on that journal code, no journal entries with that code are retrieved.

The single value *ALLFILE can be specified on the list of two values.

***ALLFILE:** The search for the journaled entry being retrieved is not limited by a specified file name.

Element 1: Physical File Name

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

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

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

library-name: Specify the name of the library to be searched.

***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 retrieved. If *ALL is specified and the user does not have the required authority to all of the files, an error occurs, and the command ends.

physical-file-name: Specify the name of the physical database file for which a journal entry is retrieved.

Element 2: Member Name

***FIRST:** An entry for the first member in the file is retrieved.

***ALL:** Journal entries for currently existing members in the file are retrieved.

member-name: Specify the name of the member for which a journal entry is retrieved.

If *ALL is specified for the file-name value, this member name is used for all applicable files in the library. For example, if library-name/*ALL *FIRST is specified on the FILE parameter, the journal entries of the first members of all applicable files in the specified library are retrieved.

If the specified physical file does not exist on the system, specify either *ALL or a specific file member name.

RCVRNG

Specifies the qualified names of the starting (first) and ending (last) journal receivers used in the search for a journal entry to be retrieved. The system starts the search with the starting journal receiver (as specified by the first value) and proceeds through the receiver chain until the ending journal receiver (as specified by the second value) is processed.

If dual receivers (pairs of receivers added or removed at the same time) are used at any time, the first of the receivers is used when chaining through the receivers. The Work with Journal Attributes (WRKJRNA) command can be used to display the order of the receivers in the receiver chain. If any problem is found in the receiver chain before the search operation begins, such as damaged or off-line receivers, the system attempts to use the second of the dual receivers. If the second of the receivers is damaged or off-line, or if a problem is found during the operation, the operation ends.

If SEARCH(*ASCEND) is specified, journal receivers must be specified in the order of oldest to newest. If SEARCH(*DESCEND) is specified, journal receivers must be specified in the order of newest to oldest.

***CURRENT:** The journal receiver that is currently attached when starting to retrieve journal entries is used.

***CURCHAIN:** The journal receiver chain that includes the journal receiver that is currently attached when starting to retrieve journal entries is used. This receiver chain does not cross a break in the chain. If there is a break in the chain, the receiver range is from the most recent break in the chain through the receiver that is attached when starting to retrieve journal entries.

Element 1: Starting Journal Receiver

The name of the starting journal receiver can be qualified by one of the following library values:

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

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

library-name: Specify the name of the library to be searched.

starting-journal-receiver: Specify the name of the first journal receiver that contains journal entries to be retrieved.

Element 2: Ending Journal Receiver

***CURRENT:** The journal receiver that is currently attached when starting to retrieve journal entries is used.

The name of the ending journal receiver can be qualified by one of the following library values:

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

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

library-name: Specify the name of the library to be searched.

ending-journal-receiver: Specify the name of the last journal receiver containing journal entries to be searched. If the end of the receiver chain is reached before a receiver of this name is found, an error message is sent and no journal entry is retrieved.

Note:

If the maximum number of receivers in the range is larger than 256, an error message is sent and no journal entry is retrieved.

FROMENT

Specifies the first journal entry considered for retrieval.

***FIRST:** The first journal entry in the specified journal receiver range is the first entry considered for retrieval. If SEARCH(*DESCEND) is specified, FROMENT(*FIRST) is valid only if TOENT(*FIRST) is also specified.

***LAST:** The last journal entry in the specified journal receiver range is the first entry considered for retrieval. If SEARCH(*ASCEND) is specified, FROMENT(*LAST) is valid only if TOENT(*LAST) is also specified.

starting-sequence-number: Specify the sequence number of the journal entry that is the first entry considered for retrieval.

FROMTIME

Specifies the date and time of the first journal entry considered for retrieval. The first journal entry found with the specified date and time or the next later journal entry is the starting point for the search.

starting-date: Specify the starting date using the format defined by the system values QDATFMT and, if separators are used, QDATSEP.

starting-time: Specify the starting time. 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.

TOENT

Specifies the last journal entry considered for retrieval.

***LAST**: The search continues until the last journal entry in the specified journal receiver range is processed. If SEARCH(*DESCEND) is specified, TOENT(*LAST) is valid only if FROMENT(*LAST) is also specified.

***FIRST**: The search continues until the first journal entry in the specified journal receiver range is processed. If SEARCH(*ASCEND) is specified, TOENT(*FIRST) is only valid if FROMENT(*FIRST) is also specified.

ending-sequence-number: Specify the sequence number of the journal entry that is the final entry considered for retrieval.

Note:

The values specified for the FROMENT and TOENT parameter can be the same (for example, FROMENT(234) and TOENT(234) can be specified).

TOTIME

Specifies the date and time of the last entry considered for retrieval. The first journal entry found with the specified date and time, or the latest earlier journal entry is the ending point for the search.

ending-date: Specify the ending date using the format defined by the system values QDATFMT and, if separators are used, QDATSEP.

ending-time: Specify the ending time. See the FROMTIME parameter for a description of time formats.

SEARCH

Specifies the order in which the journal entries are searched to retrieve an entry.

***ASCEND**: The journal entries are searched in ascending order (from the oldest entry to the newest entry).

***DESCEND**: The journal entries are searched in descending order (from the newest entry to the oldest entry).

JRNCDE

Specifies the journal codes of the journal entries being considered for retrieval.

***ALL**: The search for the entry to retrieve is not limited to a specified journal code.

***CTL:** The entries considered for retrieval are the journal entries created to control the journal functions. The journal codes for these are J and F.

Element 1: Journal Code Value

journal-code: Specify the journal code to which journal entries are limited. Only journal entries with the specified journal code are retrieved. A list of journal codes that can be specified is provided in the Journal management article in the Information Center .

Element 2: Journal Code Selection

***ALLSLT:** The journal entries with the specified journal code are retrieved only if all other selection parameters are satisfied.

***IGNFILSLT:** The journal entries having the specified journal code are retrieved only if all selection parameters, except the FILE parameter, are satisfied.

Note: This value is not valid for journal code D, F or R.

ENTTYP

Specifies whether to limit the journal entries retrieved to those of a specified journal entry type.

***ALL:** The search for the entry to retrieve is not limited to a specified entry type.

***RCD:** Only entries that have an entry type for record-level operations are retrieved. The following entry types are valid: BR, DL, DR, IL, PT, PX, UB, UP, and UR.

entry-type: Specify the entry type that limits the search for the entry journal entries to retrieve. Only journal entries that contain the specified entry type are considered for retrieval. Up to 300 valid entry types can be specified. A list of valid entry types is in the Journal management article in the Information Center .

JOB

Specifies whether the journal entries being retrieved are limited to the journal entries for a specified job. Only journal entries for the specified job are considered for retrieval. If no job name is given, all of the journal entries that contain the simple name of the job are considered for retrieval.

A job identifier is a special value or a qualified name with up to three elements. For example:

```
*ALL
*
job-name
user-name/job-name
job-number/user-name/job-name
```

More information on this parameter is in Commonly used parameters.

***ALL:** The search is not limited to entries for a specified job.

***:** The search is limited to entries for the current job. Only entries for the requesting job are considered for retrieval.

job-name: Specify the name of the job whose journal entries are being considered for retrieval.

user-name: Specify the name of the user of the job whose journal entries are being considered for retrieval.

job-number: Specify the number of the job whose journal entries are being considered for retrieval.

PGM

Specifies the name of a program whose journal entries are considered for retrieval.

***ALL:** The search is not limited to entries for a specified program.

program-name: Specify the name of the program whose journal entries are considered for retrieval. Only journal entries for this program are considered for retrieval.

USRPRF

Specifies that the journal entries to be considered for retrieval are limited to the journal entries for a specified user profile name.

***ALL:** The retrieval is not limited to entries for a specified user profile.

user-name: Specify the name of the user profile whose journal entries are to be considered for retrieval. Only changes for this user profile that were journaled are considered for retrieval.

CMTCYCID

Specifies that the journal entries considered for retrieval are limited to the journal entries that contain the specified commit cycle identifier.

***ALL:** The search is not limited to entries for a specified commit cycle identifier.

commit-cycle-identifier: Specify the commit cycle identifier that limits the search for the entry to retrieve. Only journal entries that contain the commit cycle identifier are considered for retrieval.

DEPENT

Specifies whether to retrieve the journal entries recording actions

- that occur as a result of a trigger program
- on records that are part of a referential constraint
- that will be ignored during an Apply Journaled Changes (APYJRNCHG) or Remove Journaled Changes (RMVJRNCHG) operation.

***ALL:** The journal entries relating to trigger programs, referential constraints and the entries which will be ignored by an Apply or Remove Journaled Changes operations are retrieved.

***NONE:** The journal entries relating to trigger programs, referential constraints and the entries which will be ignored by an Apply or Remove Journaled Changes operations are not retrieved.

ENTFMT

Specifies the format of the journal entries being retrieved. For a description of what is represented by each of the fields in the journal entry, see the Journal management article in the Information Center .

Lists showing detailed information on the format of the retrieved journal entries are in the RTNJRNE parameter description.

***TYPE1:** The retrieved journal entries are formatted to include the minimum information that can be specified. The retrieved journal entries include the information in the format listed in the RTNJRNE parameter description for ENTFMT(*TYPE1).

***TYPE2:** The retrieved journal entries include the information returned when ENTFMT(*TYPE1) is specified, the user profile field giving the name of the user who logged the retrieved journal entries, and the name of the system on which the entry was sent.

***TYPE3:** The retrieved journal entries include the information returned when ENTFMT(*TYPE2) is specified, and the null value indicators.

***TYPE4:** The retrieved journal entries include the information returned when ENTFMT(*TYPE3) is specified, the journal identifier, the physical file trigger indicator, and the referential constraint indicator.

➤ ***TYPE5:** The retrieved journal entries include the information returned when OUTFILFMT(*TYPE4) is specified, in addition to the program library name, the program library ASP device name, the program library ASP number, the system sequence number, the logical unit of work, the transaction identifier, the thread identifier, the remote address, the address family, the

remote port, the arm number, the receiver name, the receiver library name, the receiver library ASP number and the receiver library ASP device name. <<

NULLINDLEN

Specifies the length, in bytes, used for the null value indicators portion of the retrieved entry. This parameter is valid only if >> ENTFMT(*TYPE1) or ENTFMT(*TYPE2) << is not specified. Valid values range from 1 to 8000.

If the retrieved journal entry has fewer null value indicators than the specified field length, the trailing bytes in the null value indicators field will be set to 'F0'X.

INCENT

Specifies whether only the confirmed or both the confirmed and unconfirmed, journal entries are retrieved. This parameter only applies when converting journal entries for output from a remote journal.

Confirmed entries are those journal entries which have been sent to this remote journal and the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is known.

Unconfirmed entries are those journal entries which have been sent to this remote journal, but the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is not known, or the object name information for those journal entries is not yet known to the remote journal. Unconfirmed journal entries can only exist within the attached receiver of a remote journal. This only applies if synchronous delivery mode is being used for a particular remote journal.

***CONFIRMED:** Only those journal entries which have been confirmed are retrieved.

***ALL:** All confirmed and unconfirmed journal entries are retrieved.

RTNSEQNBR

Specifies the name of the CL decimal variable in the program into which the journal entry sequence number of the retrieved journal entry is copied. If a CL variable name is not specified, the journal entry sequence number is not copied into the program. The specified variable must be a decimal variable that has a length of ten positions with no decimal positions.

RTNJRNCD

Specifies the name of the CL character variable in the program into which the journal code of the retrieved journal entry is copied. If a CL variable name is not specified, the journal code of the retrieved journal entry is not copied into the program. The specified variable must be a character variable with a minimum length of 1 character. If the length of the variable is longer than 1 character, it is padded on the right with blanks.

RTNENTTYP

Specifies the name of the CL character variable in the program into which the entry type of the retrieved journal entry is copied. If a CL variable name is not specified, the entry type of the retrieved journal entry is not copied into the program. The specified variable must be a character variable with a minimum length of 2 characters. If the length of the variable is longer than 2 characters, it is padded on the right with blanks.

RTNRCV

Specifies the name of the CL character variable in the program into which the journal receiver name from where the returned journal entry was retrieved is copied. If the CL variable name is not specified, the journal receiver name is not copied into the program. The specified variable must be a character variable with a minimum length of 10 characters. If the length of the variable is longer than 10 characters, it is padded on the right with blanks.

RTNRCVLIB

Specifies the name of the CL character variable in the program into which journal receiver library name identified by the returned journal entry was retrieved is copied. If the CL variable name is not specified, the journal receiver library name is not copied into the program. The specified

variable must be a character variable with a minimum length of 10 characters. If the length of the variable is longer than 10 characters, it is padded on the right with blanks.

RTNJRNE

Specifies the name of the CL character variable in the program into which the retrieved journal entry is copied. If a CL variable name is not specified, the retrieved journal entry is not copied into the program. The specified variable must be a character variable. If the retrieved journal entry is longer than the variable's field length, the entry is truncated. If the entry is shorter, it is padded on the right with blanks.

The journal entry can be retrieved in one of  the following  possible formats.

The following lists show detailed information on the format of the retrieved journal entries.

If ENTFRMT(*TYPE1) is specified, then the format of the fields in the retrieved entry is as follows:

Field Name	Field Attributes
ENTRY LENGTH	TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER	TYPE(*DEC) LEN(10 0)
JOURNAL CODE	TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE	TYPE(*CHAR) LEN(2)
DATE	TYPE(*CHAR) LEN(6)
TIME	TYPE(*DEC) LEN(6 0)
JOB NAME	TYPE(*CHAR) LEN(10)
USER NAME	TYPE(*CHAR) LEN(10)
JOB NUMBER	TYPE(*DEC) LEN(6 0)
PROGRAM NAME	TYPE(*CHAR) LEN(10)
OBJECT NAME	TYPE(*CHAR) LEN(10)
OBJECT LIBRARY	TYPE(*CHAR) LEN(10)
MEMBER NAME	TYPE(*CHAR) LEN(10)
COUNT/RRN	TYPE(*DEC) LEN(10 0)
FLAG	TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID	TYPE(*DEC) LEN(10 0)
INCOMPLETE DATA	TYPE(*CHAR) LEN(1)
MINIMIZED ENTRY DATA	TYPE(*CHAR) LEN(1)
RESERVED	TYPE(*CHAR) LEN(6)
ENTRY-SPECIFIC DATA	TYPE(*CHAR) LEN(up to 9844)

If ENTFRMT(*TYPE2) is specified, the format of the fields in the retrieved entry is as follows:

Field Name	Field Attributes
ENTRY LENGTH	TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER	TYPE(*DEC) LEN(10 0)
JOURNAL CODE	TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE	TYPE(*CHAR) LEN(2)
DATE	TYPE(*CHAR) LEN(6)
TIME	TYPE(*DEC) LEN(6 0)
JOB NAME	TYPE(*CHAR) LEN(10)
USER NAME	TYPE(*CHAR) LEN(10)
JOB NUMBER	TYPE(*DEC) LEN(6 0)
PROGRAM NAME	TYPE(*CHAR) LEN(10)
OBJECT NAME	TYPE(*CHAR) LEN(10)
OBJECT LIBRARY	TYPE(*CHAR) LEN(10)
MEMBER NAME	TYPE(*CHAR) LEN(10)
COUNT/RRN	TYPE(*DEC) LEN(10 0)
FLAG	TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID	TYPE(*DEC) LEN(10 0)
USER PROFILE	TYPE(*CHAR) LEN(10)
SYSTEM NAME	TYPE(*CHAR) LEN(8)
INCOMPLETE DATA	TYPE(*CHAR) LEN(1)
MINIMIZED ENTRY DATA	TYPE(*CHAR) LEN(1)
RESERVED	TYPE(*CHAR) LEN(18)
ENTRY-SPECIFIC DATA	TYPE(*CHAR) LEN(up to 9844)

If ENTFRMT(*TYPE3) is specified and a value is specified on the NULLINDLEN parameter, the format of the retrieved journal entry is as follows:

Field Name	Field Attributes
ENTRY LENGTH	TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER	TYPE(*DEC) LEN(10 0)
JOURNAL CODE	TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE	TYPE(*CHAR) LEN(2)
TIMESTAMP	TYPE(*TIMESTAMP) LEN(26)
JOB NAME	TYPE(*CHAR) LEN(10)
USER NAME	TYPE(*CHAR) LEN(10)
JOB NUMBER	TYPE(*DEC) LEN(6 0)
PROGRAM NAME	TYPE(*CHAR) LEN(10)
OBJECT NAME	TYPE(*CHAR) LEN(10)
OBJECT LIBRARY	TYPE(*CHAR) LEN(10)
MEMBER NAME	TYPE(*CHAR) LEN(10)
COUNT/RRN	TYPE(*DEC) LEN(10 0)
FLAG	TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID	TYPE(*DEC) LEN(10)
USER PROFILE	TYPE(*CHAR) LEN(10)
SYSTEM NAME	TYPE(*CHAR) LEN(8)
NULL VALUE INDICATORS	TYPE(*CHAR) field-length ¹
ENTRY-SPECIFIC DATA	TYPE(*CHAR) ((up to 9850) minus (field length)) ²

Notes:

- 1 The length of this field is the length specified on the NULLINDLEN parameter.
- 2 The length of this portion of the entry depends on the length specified on the RTNJRNE parameter and the length specified on the NULLINDLEN parameter.

Note: If ENT_FMT(*TYPE3) is specified, 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 article in the Information Center for more information on the incomplete data indicator, minimized entry specific data indicator, and these journal entries.

If ENT_FMT(*TYPE4) is specified and a value is specified on the NULLINDLEN parameter, the format of the retrieved journal entry is as follows:

Field Name	Field Attributes
ENTRY LENGTH	TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER	TYPE(*DEC) LEN(10 0)
JOURNAL CODE	TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE	TYPE(*CHAR) LEN(2)
TIMESTAMP	TYPE(*TIMESTAMP) LEN(26)
JOB NAME	TYPE(*CHAR) LEN(10)
USER NAME	TYPE(*CHAR) LEN(10)
JOB NUMBER	TYPE(*DEC) LEN(6 0)
PROGRAM NAME	TYPE(*CHAR) LEN(10)
OBJECT NAME	TYPE(*CHAR) LEN(10)
OBJECT LIBRARY	TYPE(*CHAR) LEN(10)
MEMBER NAME	TYPE(*CHAR) LEN(10)
COUNT/RRN	TYPE(*DEC) LEN(10 0)
FLAG	TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID	TYPE(*DEC) LEN(10)
USER PROFILE	TYPE(*CHAR) LEN(10)
SYSTEM NAME	TYPE(*CHAR) LEN(8)
JOURNAL IDENTIFIER	TYPE(*CHAR) LEN(10)
REF CONSTRAINT	TYPE(*CHAR) LEN(1)
TRIGGER	TYPE(*CHAR) LEN(1)
INCOMPLETE DATA	TYPE(*CHAR) LEN(1)
IGNORE DURING	TYPE(*CHAR) LEN(1)
APYJRNCHG OR RMVJRNCHG	

MINIMIZED ENTRY DATA	TYPE(*CHAR) LEN(1)
RESERVED	TYPE(*CHAR) LEN(5)
NULL VALUE INDICATORS	TYPE(*CHAR) field-length ¹
ENTRY-SPECIFIC DATA	TYPE(*CHAR) ((up to 9830) minus (field length)) ²

Notes:

- ¹ The length of this field is the length specified on the NULLINDLEN parameter.
- ² The length of this portion of the entry depends on the length specified on the RTNJRNE parameter and the length specified on the NULLINDLEN parameter.

➤ If ENTFMT(*TYPE5) is specified, and a value is specified on the NULLINDLEN parameter, the format of the retrieved journal entry is as follows:

Field Name	Field Attributes
ENTRY LENGTH	TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER	TYPE(*CHAR) LEN(20)
JOURNAL CODE	TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE	TYPE(*CHAR) LEN(2)
TIMESTAMP	TYPE(*TIMESTAMP) LEN(26)
JOB NAME	TYPE(*CHAR) LEN(10)
USER NAME	TYPE(*CHAR) LEN(10)
JOB NUMBER	TYPE(*DEC) LEN(6 0)
PROGRAM NAME	TYPE(*CHAR) LEN(10)
PROGRAM LIBRARY NAME	TYPE(*CHAR) LEN(10)
PROGRAM ASP DEVICE NAME	TYPE(*CHAR) LEN(10)
PROGRAM ASP NUMBER	TYPE(*DEC) LEN(5 0)
OBJECT NAME	TYPE(*CHAR) LEN(10)
OBJECT LIBRARY	TYPE(*CHAR) LEN(10)
MEMBER NAME	TYPE(*CHAR) LEN(10)
COUNT/RRN	TYPE(*CHAR) LEN(20)
FLAG	TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID	TYPE(*CHAR) LEN(20)
USER PROFILE	TYPE(*CHAR) LEN(10)
SYSTEM NAME	TYPE(*CHAR) LEN(8)
JOURNAL IDENTIFIER	TYPE(*CHAR) LEN(10)
REF CONSTRAINT	TYPE(*CHAR) LEN(1)
TRIGGER	TYPE(*CHAR) LEN(1)
INCOMPLETE DATA	TYPE(*CHAR) LEN(1)
IGNORE DURING	TYPE(*CHAR) LEN(1)
APYJRNCHG OR RMVJRNCHG	
MINIMIZED ENTRY DATA	TYPE(*CHAR) LEN(1)
OBJECT INDICATOR	TYPE(*CHAR) LEN(1)
SYSTEM SEQUENCE NUMBER	TYPE(*CHAR) LEN(20)
RECEIVER NAME	TYPE(*CHAR) LEN(10)
RECEIVER LIBRARY NAME	TYPE(*CHAR) LEN(10)
RECEIVER ASP DEVICE NAME	TYPE(*CHAR) LEN(10)
RECEIVER ASP NUMBER	TYPE(*DEC) LEN(5 0)
ARM NUMBER	TYPE(*DEC) LEN(5 0)
THREAD IDENTIFIER	TYPE(*CHAR) LEN(8)
THREAD IDENTIFIER HEX	TYPE(*CHAR) LEN(16)
ADDRESS FAMILY	TYPE(*CHAR) LEN(1)
REMOTE PORT	TYPE(*DEC) LEN(5 0)
REMOTE ADDRESS	TYPE(*CHAR) LEN(46)
LOGICAL UNIT OF WORK	TYPE(*CHAR) LEN(39)
TRANSACTION IDENTIFIER	TYPE(*CHAR) LEN(140)
RESERVED	TYPE(*CHAR) LEN(20)
NULL VALUE INDICATORS	TYPE(*CHAR) field-length ¹
ENTRY-SPECIFIC DATA	TYPE(*CHAR) ((up to 9443) minus (field length)) ²

Notes:

- 1 The length of this field is the length specified on the NULLINDLEN parameter.
- 2 The length of this portion of the entry depends on the length specified on the RTNJRNNE parameter and the length specified on the NULLINDLEN parameter. <<

Examples for RTVJRNE**Example 1**

Assume the following variables are specified:

```
DCL &SEQ          TYPE(*DEC)  LEN(10 0)
DCL &JRNENT       TYPE(*CHAR)  LEN(200)
DCL &RCVNAME      TYPE(*CHAR)  LEN(10)
DCL &RCVLIB       TYPE(*CHAR)  LEN(10)
```

and this command is run:

```
RTVJRNE  JRN(MYLIB/JRNA)  ENTYP(PR)
         RTNSEQNBR(&SEQ#)  RTNJRNNE(&JRNENT)
```

Since no starting journal entry is specified in this command, the first entry from the journal receiver that is currently attached to the journal JRNA in the library MYLIB, when starting to retrieve entries, is considered for retrieval. The first entry in any receiver is always an identifier for the previously-attached receiver. This first receiver entry is known as a type PR entry, and it contains the name of the previously attached receiver in its entry-specific data. The PR entry is the first entry in ascending order in the currently attached receiver; when it is found, the entry is placed into a CL variable named &JRNENT.

Then Change Variable (CHGVAR) can be used to separate the name and library of the previous journal receiver, found in the entry specific data, as follows:

```
CHGVAR &RCVNAME  (%SST(&JRNENT 126 10))
CHGVAR &RCVLIB   (%SST(&JRNENT 136 10))
```

Example 2

Assume the following variables are specified:

```
DCL &ENTNO        TYPE(*DEC)  LEN(10 0)
DCL &JCODE        TYPE(*CHAR)  LEN(1)
DCL &ETYP        TYPE(*CHAR)  LEN(2)
DCL &RCVNAME      TYPE(*CHAR)  LEN(10)
DCL &RCVLIB       TYPE(*CHAR)  LEN(10)
DCL &JENTRY       TYPE(*CHAR)  LEN(205)
```

and this command is run:

```
RTVJRNE  JRN(MYLIB/JRNLA)  FILE(LIB1/A MBR3)
         RCVRNG(RCVLIB/RCV30 RCVLIB/RCV27)
         ORDER(*DESCEND)  JRNCDE(R)  ENTYP(UP DL)
         JOB(000666/QPGMR/PRESTR)  PGM(WAKEUP)
         USRPRF(MAC7)  ENTFMT(*TYPE2)
         RTNSEQNBR(&ENTNO)  RTNJRNRCDE(&JCODE)
         RTNENTYP(&ETYP)  RTNRCV(&RCVNAME)
         RTNRCVLIB(&RCVLIB)  RTNJRNNE(&JENTRY)
```

This command gets a journal entry, searching in descending order the journal receiver chain from receiver RCV30 in library RCVLIB to receiver RCV27 in library RCVLIB, journaled through journal JRNLA in library MYLIB, and copies the entry into the specified CL variables. The retrieved entry is an UPDATE or DELETE entry with journal code R from member MBR3 in file A in library LIB1, created in job

000666/QPGMR/PRESTR in program WAKEUP by user profile MAC7. The retrieved journal entry includes the user profile field. The sequence number of the retrieved entry is copied into CL variable &ENTNO. The journal code of the retrieved entry is copied into CL variable &JCODE. The entry type of the retrieved entry is copied into CL variable &ETYPE. The name of the journal receiver from which the returned entry was retrieved is copied into &RCVNAME. The library name of the journal receiver from which the returned entry was retrieved is copied into &RCVLIB.

Error messages for RTVJRNE

***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).

CPF7060

File &1 member &5 never journaled to journal &3.

CPF7061

Conversion of journal entries failed.

CPF7062

No entries converted or received from journal &1.

CPF7065

Entry type (ENTTYP) not valid for journal code (JRNCDE).

CPF7072

Retrieval of journal entry failed.

CPF7073

No entry retrieved from journal &1 in &2.

CPF7074

RCVRNG for specified SEARCH not valid.

CPF708D

Journal receiver found logically damaged.

CPF709C

JOB, PGM, and USRPRF not valid for receiver range.

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.

RTVLIBD (Retrieve Library Description) Command Description

RTVLIBD Command syntax diagram

Purpose

The Retrieve Library Description (RTVLIBD) command is used in a CL program or REXX procedure to retrieve the description of a library.

The CL prompt for this command lists the minimum length for each variable next to the parameters being retrieved. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

1. The user cannot retrieve the attributes of a library for which they have *EXCLUDE authority.

Required Parameter

LIB Specifies the name of the library for which information is being retrieved. If a variable is specified, it must be 10 characters in length and include a library name.

Optional Parameters

TYPE Specifies the name of a 10-character CL variable that is used to retrieve the type of the library.

The value that can be returned is PROD or TEST.

CRTAUT

Specifies the name of a 10-character CL variable that is used to return the create authority value of the library.

The value that can be returned is *SYSVAL, *CHANGE, *ALL, *USE, *EXCLUDE, or the name of an authorization list.

ASP Specifies the name of a decimal variable that is used to retrieve the number of the auxiliary storage pool (ASP) from which the system allocates storage for the library. In CL programs, this must be a decimal variable length of (2 0). ➤ This variable will contain an ASP number up to 32. If

an ASP number is larger than 32, -1 is returned in this variable. Parameter ASPDEV should be used to return the ASP device name when the ASP number is larger than 32. << The following values can be returned:

- 1 The >> library << is in the system auxiliary storage pool.
- 2-32 The >> library is in a basic user auxiliary storage pool. <<
- >>-1 The library is in a primary or secondary auxiliary storage pool (ASP) with an ASP number larger than 32. The ASP device name for the primary or secondary ASP can be retrieved with the ASPDEV parameter. <<

>> ASPDEV

Specifies the name of a 10-character CL variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the library. One of the following special values can be returned:

*N The name of the ASP device can not be determined.

*SYSBAS

The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32). <<

CRTOBJAUD

Specifies the name of a 10-character variable used to return the auditing value of the library. The values that can be retrieved include *SYSVAL, *NONE, *USRPRF, *CHANGE, and *ALL. See the CRTOBJAUD parameter on the Create Library (CRTLIB) command for more information.

TEXT Specifies the name of a 50-character CL variable that is used to return the text description to a library.

Example for RTVLIBD

```
CRTLIB LIB(TESTLIB) CRTAUT(*ALL)
TEXT('John Smith library')
```

This command creates John Smith's library.

```
DCL &CRTAUT *CHAR 10
RTVLIBD LIB(TESTLIB)
CRTAUT(&CRTAUT)
```

When user Smith calls a CL program containing the above, the following values are returned:

```
&CRTAUT ~*ALL
```

Error messages for RTVLIBD

*ESCAPE Messages

>> CPF8ED

Device description &1 not correct for operation. <<

CPF210E

Library &1 not available.

CPF2115

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

CPF2150

Object information function failed.

CPF2151

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

» CPF980B

Object &1 in library &2 not available. «

CPF9810

Library &1 not found.

CPF9820

Not authorized to use library &1.

CPF9830

Cannot assign library &1.

RTVMBRD (Retrieve Member Description) Command Description

RTVMBRD Command syntax diagram

Purpose

The Retrieve Member Description (RTVMBRD) command is used in a CL program or REXX procedure to retrieve (return) the member-level information (in CL variables) from a database file.

The values are returned (copied) to the specified CL variables. The following kinds of member information can be retrieved:

- The library name.
- The member name.
- The file attribute.
- The file type.
- The source type.
- The source date.
- The date created.
- The expiration date.
- The member text.
- The number of nondeleted records.
- The number of deleted records.
- The open data path status (shared or not shared).
- The data space size.
- The access path size.
- The date changed.
- The date saved.
- The date restored.
- The number of data members.
- The last date used.
- The days count used.
- The date and days count was reset.

Required Parameter

FILE Specifies the qualified name of the file that contains the member description being retrieved.

Note: *OBJOPR authority to the file and *READ authority to the library are required for the member description being retrieved.

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

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

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

library-name: Specify the name of the library to be searched.

file-name: Specify the name of the file.

Optional Parameters

MBR Specifies the member whose description is retrieved.

***FIRST:** The first member in the database file is used.

***LAST:** The last member of the specified physical file is retrieved.

Element 1: Member to Retrieve

***FIRSTMBR:** The first member in a list sorted by name is retrieved.

***LASTMBR:** The last member in a list sorted by name is retrieved.

member-name: Specify the name of the reference member. The reference member is a base from which the *NEXT, *PRV, or *SAME member is retrieved.

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

Element 2: Operation to Perform on Member

***SAME:** The reference member is retrieved.

***NEXT:** The member immediately after the reference member is retrieved.

***PRV:** The member immediately before the reference member is retrieved.

RTNSYSTEM

Specifies the name of a 4-character CL variable used to retrieve an indication of the system from which the description is retrieved. The possible values are *LCL (file found on the local system) and *RMT (file found on remote system).

RTNLIB

Specifies the name of a 10-character CL variable used to retrieve the name of the library where the specified file is located.

RTNMBR

Specifies the name of a 10-character CL variable used to retrieve the name of the member whose description is being retrieved.

FILEATR

Specifies the name of a 3-character CL variable used to retrieve the file attribute. The possible values are *PF (retrieves the attribute for a physical file member) and *LF (retrieves the attribute for a logical file member).

FILETYPE

Specifies the name of a 5-character CL variable used to retrieve the file type. The possible values are *DATA (retrieves the file type for a data member) and *SRC (retrieves the file type for a source member).

SRCTYPE

Specifies the name of a 10-character CL variable used to retrieve the source member type (if this is a source member). Blanks are returned if this is not a source member.

SRCCHGDATE

Specifies the name of a 13-character CL variable used to retrieve the century, date, and time the last source file member was changed. The format is CYYMMDDHHMMSS where C=Century (0=1940 through 1999 and 1=2000 through 2039), Y=Year, M=Month, D=Day, H=Hour, M=Minutes, and S=Seconds. Blanks are returned if no date is available or if the date of remote non-iSeries 400 or non-System/38 files are retrieved.

CRDATE

Specifies the name of a 13-character CL variable used to retrieve the member creation century, date, and time. The format is CYYMMDDHHMMSS where C=Century (0=1940 through 1999 and 1=2000 through 2039), Y=Year, M=Month, D=Day, H=Hour, M=Minutes, and S=Seconds.

EXPDATE

Specifies the name of a 7-character CL variable used to retrieve the member expiration century, date, and time. The format is CYYMMDD where C=Century (0=1940 through 1999 and 1=2000 through 2039), Y=Year, M=Month, and D=Day. *NONE is returned if no date is available.

TEXT Specifies the name of a 50-character CL variable used to retrieve the text description of the file member. More information on this parameter is in Commonly used parameters.

NBRCURRCD

Specifies the name of a 10-position decimal CL variable used to retrieve the current number of nondeleted records in the member. If the member is a keyed logical member, the number of index entries is returned. For nonkeyed logical members, the number of nondeleted records in the based-on physical file member is returned.

NBRDLTRCD

Specifies the name of a 10-position decimal CL variable used to retrieve the current number of deleted records in this member. Zero (0) is returned for keyed logical files. For nonkeyed logical files, the number of nondeleted records in the based-on physical file member is returned.

SHARE

Specifies whether the open data path (ODP) for the member description is shared with other programs in the 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 Programming topic in the Information Center.

DTASPCSI

Specifies the name of a 15-position decimal CL variable used to retrieve the data space size (in bytes) of this member. Zero (0) is returned if this is a logical member.

ACCPHSIZ

Specifies the name of a 12-position decimal CL variable used to retrieve the access path size (in bytes) for this member. Zero (0) is returned if the member is nonkeyed. Remote non-iSeries 400 and non-System 38 files return a value of zero (0).

CHGDATE

Specifies the name of a 13-character CL variable used to retrieve the member change century, date, and time. The format is CYYMMDDHHMMSS where C=Century (0=1940 through 1999 and 1=2000 through 2039), Y=Year, M=Month, D=Day, H=Hour, M=Minutes, and S=Seconds.

SAVDATE

Specifies the name of a 13-character CL variable used to retrieve the member save century, date, and time. The format is CYYMMDDHHMMSS where C=Century (0=1940 through 1999 and 1=2000 through 2039), Y=Year, M=Month, D=Day, H=Hour, M=Minutes, and S=Seconds. Blanks are returned if no date is available. Remote non-iSeries 400 and non-System 38 files return blanks.

RSTDATE

Specifies the name of a 13-character CL variable used to retrieve the member restore century, date, and time. The format is CYYMMDDHHMMSS where C=Century (0=1940 through 1999 and 1=2000 through 2039), Y=Year, M=Month, D=Day, H=Hour, M=Minutes, and S=Seconds. Blanks are returned if there is no date available.

NBRDTAMBR

Specifies the name of a 2-position decimal CL variable used to retrieve the number of data members for this logical member. If the member is a physical member, zero (0) is returned.

USEDATE

Specifies the name of a 7-character CL variable that is used to retrieve the date the object was used last. The date will be returned in the form CYYMMDD where C=Century (0=1940 through 1999 and 1=2000 through 2039), Y=Year, M=Month, and D=Day. If the object does not have a last-used date, blanks will be returned.

USECOUNT

Specifies the name of a 5 position decimal CL variable used to return the number of days the object has been used. If the object does not have a last used date, 0 will be returned.

RESETDATE

Specifies the name of a 7-character CL variable that is used to return the date the days-used count was last reset to 0. The date will be returned in the form CYYMMDD where C=Century (0=1940 through 1999 and 1=2000 through 2039), Y=Year, M=Month, and D=Day. If the days used count has not been reset, blanks will be returned.

Examples for RTVMBRD

Assume the user has a file named MYFILE in library MYLIB (which is the current library) with members QMEMBER, BMEMBER, ZMEMBER, and JMEMBER (created in that order).

Also assume the following variables are specified in the CL program:

```
DCL &LIB      TYPE(*CHAR) LEN(10)
DCL &MBR      TYPE(*CHAR) LEN(10)
DCL &SYS      TYPE(*CHAR) LEN(4)
DCL &MTYPE    TYPE(*CHAR) LEN(5)
DCL &CRTDATE  TYPE(*CHAR) LEN(13)
DCL &CHGDATE  TYPE(*CHAR) LEN(13)
DCL &TEXT     TYPE(*CHAR) LEN(50)
DCL &NBRRCN  TYPE(*DEC)  LEN(10 0)
DCL &SIZE     TYPE(*DEC)  LEN(10 0)
```

This command is run:

```
RTVMBRD FILE(*CURLIB/MYFILE) MBR(BMEMBER *SAME)
RTNLIB(&LIB) RTNSYSTEM(&SYS) RTNMBR(&MBR)
FILEATR(&MTYPE) CRTDATE(&CRTDATE) TEXT(&TEXT)
NBRCURRCD(&NBRRCD) DTASPCSIZE(&SIZE)
```

The requested information is placed in the CL variables as follows:

- The current library name (MYLIB) is placed in the CL variable named &LIB.
- The system on which MYFILE was found is placed in the CL variable named &SYS. (*LCL means the file was found on the local system, and *RMT means the file was found on a remote system.)
- The member name (BMEMBER) is placed in the CL variable named &MBR.
- The file attribute of MYFILE is placed in the CL variable named &MTYPE. (*DATA means the member is a data member, and *SRC means the file is a source member.)
- The creation date of BMEMBER is placed in the CL variable named &CRTDATE.
- The text associated with BMEMBER is placed in the CL variable called &TEXT.
- The current number of records in BMEMBER is placed in the CL variable called &NBRRCD.
- The size of BMEMBER's data space (in bytes) is placed in the CL variable called &SIZE.

This command is run next:

```
RTVMBRD FILE(&LIB/MYFILE) MBR(&MBR *NEXT)
RTNMBR(&MBR) CRTDATE(&CRTDATE) TEXT(&TEXT)
NBRCURRCD(&NBRRCD) DTASPCSIZE(&SIZE)
```

Then the requested information is placed in the CL variables as follows:

- The next members name after BMEMBER (JMEMBER since the file is searched in name order) in MYFILE is placed in the CL variable named &MBR.
- The creation date of JMEMBER is placed in the CL variable named &CRTDATE.
- The text associated with JMEMBER is placed in the CL variable called &TEXT.
- The current number of records in JMEMBER is placed in the CL variable called &NBRRCD.
- The size of JMEMBER's data space (in bytes) is placed in the CL variable called &SIZE.

The file can also be searched backwards. An example is:

```
RTVMBRD FILE(*CURLIB/MYFILE) MBR(ZMEMBER *PRV)
RTNMBR(&MBR) CHGDATE(&CHGDATE) TEXT(&TEXT)
```

The requested information is placed in the CL variables as follows:

- The member name (QMEMBER since it is the member just previous to ZMEMBER in a name ordered list) is placed in the CL variable named &MBR.
- The date QMEMBER was last changed is placed in the CL variable named &CHGDATE.
- The text associated with QMEMBER is placed in the CL variable called &TEXT.

If only the first part of the member name is known, the user can do a Generic* (or partial name) search of the list of members, as follows:

```
RTVMBRD FILE(*LIBL/MYFILE) MBR(JM*) RTNMBR(&MBR)
CHGDATE(&CHGDATE) TEXT(&TEXT)
```

The requested information is placed in the CL variables as follows:

- The member name (JMEMBER since it is the first member starting with the characters JM in a name ordered list) is placed in the CL variable named &MBR.
- The date JMEMBER was last changed is placed in the CL variable named &CHGDATE.
- The text associated with JMEMBER is placed in the CL variable called &TEXT.

Error messages for RTVMBRD

***ESCAPE Messages**

CPF3018

Member &3 for file &1 in &2 not available.

CPF3019

File &1 in library &2 has no members.

CPF3027

File &1 in &2 not a database file.

CPF3038

Attributes for return variable &1 not valid.

CPF3039

Return variable &1 too small to hold result.

CPF3049

*NEXT or *PRV member does not exist.

CPF3051

File &1 in library &2 not available.

CPF325F

Conversion of the text failed.

CPF327B

File &1 information cannot be retrieved.

CPF8109

&8 damage on physical database file &4.

CPF8110

&8 damage on logical data base file &4 in &9. VLOG-&7.

CPF8111

&8 damage on member &9 file &4.

CPF9803

Cannot allocate object &2 in library &3.

CPF9806

Cannot perform function for object &2 in library &3.

CPF9810

Library &1 not found.

CPF9812

File &1 in library &2 not found.

CPF9815

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

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

RTVMSG (Retrieve Message) Command Description

RTVMSG Command syntax diagram

Purpose

The Retrieve Message (RTVMSG) command is used in a CL program or REXX procedure to retrieve a specified predefined message from a message file and to copy it into CL variables. Substitution values can be specified in the MSGDTA parameter (as a single character string containing one or more concatenated message data fields) to replace the substitution variables in the predefined message text. The program can later write the message to an output device file to be printed, for example.

The CL prompt for this command lists the minimum length for retrieved variables next to the parameters that have a minimum length. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restriction: The user of this command must have *USE authority for the message file and the library in which the message file is stored.

Required Parameters

MSGID

Specifies the message identifier of the predefined message being retrieved from the specified message file.

MSGF Specifies the qualified name of the message file that contains the predefined message being retrieved.

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

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

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

library-name: Specify the name of the library to be searched.

message-file-name: Specify the name of the message file to use.

Optional Parameters

CCSID

Specifies the coded character set identifier (CCSID) in which you want your message text returned. This applies only to text returned in the MSG and SECLVL parameters. When replacement data is substituted into the text returned in the MSG or SECLVL parameters, only the part of the replacement data that is defined as a character that can be converted (*CCHAR) is converted. The rest of the replacement data will not be converted. For more information about the *CCHAR field, see the Add Message Description (ADDMSGD) command.

***JOB:** The retrieved message description is converted to the CCSID of the job before being returned.

***HEX:** The retrieved message description is not converted before being returned.

coded-character-set-identifier: Specify the CCSID that you want your message description converted to before it is returned.

Note:


The valid values range from 1 through 65535. See the Globalization topic in the Information Center for a list of valid CCSID values. Only those CCSID values that the 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 Information Center.

MSGDTA

Specifies the substitution values that are used in the retrieved message if the predefined message contains substitution variables. Either a character string or a CL variable containing the character string can be specified. As many substitution values can be specified in the character string as there are substitution variables in the predefined message. The values, which take the place of the substitution variables defined in the message text when the message was defined, must be specified by the following rules:

- Multiple values must be concatenated to form a single character string. The length of the entire character string of concatenated substitution values cannot exceed 512 characters. The CL

Programming  book has more details.

- Multiple values must be specified in the same order in the character string as the substitution variables were defined in the FMT parameter of the ADDMSGD command.
- Each value must be specified as long as its associated variable was defined, and the specified character string must be the same length as the sum of the message data fields defined in the message description.
- For multiple values, each substitution value can be specified as a CL program constant or CL variable; that is, any combination of constants and variables can be specified if they are first concatenated into one character string and they are in the same format and sequence expected.
- The length of the substitution value should be the same length as the length defined for the substitution variables. If the substitution value length is longer than the substitution variable length, the message data is truncated. If the substitution value is shorter, it becomes a null field.
- The message data supplied by this command that corresponds to a *CCHAR type field is assumed to be in the CCSID of the job running this command unless the MDTACCSID parameter is coded. For more information on the *CCHAR type field, see the Add Message Description (ADDMSGD) command.

For more information on coding this parameter, refer to the description of the MSGDTA parameter in the SNDUSRMSG or SNDPGMMSG commands.

MSG Specifies the name of the CL character variable in the program into which the first-level message text from the retrieved message is copied. If a CL variable name is not specified, the text is not copied into the program.

The specified variable must be a character variable. If the retrieved text is longer than the variable's field length, the text is truncated. If the text is shorter, it is padded to the right with blanks. Although this is a variable-length field, most first-level message text is designed to be less than 132 characters.

MSGLEN

Specifies the name of the CL decimal variable in the program into which the total length of the text available to be retrieved is copied. The length specified is the total length after the substitution values (specified in the MSGDTA parameter) are placed in the text.

The specified variable must be a decimal variable that has a length of five digits. If a CL variable name is not specified, the length is not copied into the program.

SECLVL

Specifies the name of the CL character variable in the program into which the second-level message text of the retrieved message is copied. If a variable name is not specified, the second-level message text is not copied into the program.

If the retrieved second-level message text is longer than the variable's field length, the text is truncated. If the text is shorter, it is padded on the right with blanks. Although this is a variable length field, most second-level message text is designed to be less than 3000 characters.

SECLVLEN

Specifies the name of the CL decimal variable in the program into which the total length of the second-level message text being retrieved is copied. The length specified is the total length after the substitution values (specified in the MSGDTA parameter) are placed in the second-level message text.

The specified variable must be a decimal variable that has a length of five positions. If a CL variable name is not specified, the length is not copied into the program.

SEV

Specifies the name of the CL decimal variable into which the severity code of the retrieved message is copied. The specified variable must be a decimal variable that has a length of two positions. If a variable name is not specified, the severity code of the retrieved message is not copied into the program.

ALROPT

Specifies the name of the CL variable into which the alert option of the retrieved message is copied. The variable must be a character variable nine positions long.

LOGPRB

Specifies the name of the variable into which a Y (yes) or N (no) is returned indicating whether or not the message can be logged in the service activity log. The variable must be a character variable one position long.

TXTCCSID

Specifies the name of the CL variable, if any, used to return the coded character set identifier (CCSID) associated with the text returned by the MSG and SECLVL parameters. 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 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 is stored in, specify 65535 on the CCSID parameter. The CCSID that the message description is stored in is returned in the TXTCCSID parameter. You can also check for a conversion error by comparing the CCSID you passed in against the TXTCCSID returned. If they are not equal and they are not 65535, a conversion error occurred.

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 specified on the MDTACCSID parameter is returned. Otherwise the CCSID you wanted the text converted to is returned. In the case when there is no *CCHAR replacement data in the text, 65535 is returned. You can check for a conversion error by comparing the CCSID you passed in against the DTACCSID returned. If they are not equal and they are not 65535, a conversion error occurred.

MDTACCSID

Specifies the CCSID that the supplied message data is to be considered in. This only applies to the parts of the replacement data that are defined as *CCHAR. The rest of the replacement data will never be converted and is assumed to have a CCSID of 65535.

***JOB:** The message data supplied is assumed to be in the CCSID of the job running this command.

***HEX:** The message data supplied is assumed to be 65535 and is never converted.

coded-character-set-identifier: The message data supplied is assumed to be in the CCSID specified. Valid values range from 1 through 65535. For a list of valid CCSID values, see the Globalization topic in the Information Center.

Examples for RTVMSG

Example 1: Replacing Substitution Variables

```
RTVMSG MSGID(UIN0145) MSGF(INVN) MSG(&WORK)
      MSGDTA('any old time')
```

This command retrieves the message text of the message UIN0145 stored in the INVN message file. The retrieved text is copied into the CL variable &WORK after the substitution variables are replaced with the values *any*, *old*, and *time*. This example assumes that the substitution variables &1, &2, and &3 have been defined in the message as character variables, each 4 characters long.

Example 2: Retrieving First- and Second-Level Message Text

```
RTVMSG MSGID(UIN0150) MSGF(INV) MSG(&MSG)
      SECLVL(&SECLVL)
```

This command retrieves the first- and second-level message text of the message UIN0150, which is stored in message file INV, and moves it into the CL variables &MSG and &SECLVL.

Error messages for RTVMSG

*ESCAPE Messages

CPF2401

Not authorized to library &1.

CPF2407

Message file &1 in &2 not found.

CPF2411

Not authorized to message file &1 in &2.

CPF247E

CCSID &1 is not valid.

CPF2471

Length of field not valid.

CPF2499

Message identifier &1 not allowed.

CPF2531

Message file &1 in &2 damaged for &3.

CPF2547

Damage to message file QCPFMSG.

CPF2548

Damage to message file &1 in &2.

CPF8126

Message file &4 in &9 damaged.

CPF9830

Cannot assign library &1.

RTVNETA (Retrieve Network Attributes) Command Description

RTVNETA Command syntax diagram

Purpose

The Retrieve Network Attributes (RTVNETA) command is used in a CL program or REXX procedure to retrieve the network attributes of the system.

The values are returned (copied) to the specified CL variables.

Restriction: The attributes of the network attribute and the receiving CL variable must be compatible.

Optional Parameters

SYSNAME

Specifies the name of the CL variable that receives the current system name. Specify the name of a character variable with a minimum length of 8 characters. If the system name has fewer characters than the variable allows, the value is padded on the right with blanks.

PNDSYSNAME

Specifies the name of the CL variable that receives the pending system name. Specify the name of the character variable with a minimum length of 8 characters. The value returned is blank if no pending system name is provided. If the system name has fewer characters than the variable allows, the value is padded on the right with blanks.

LCLNETID

Specifies the name of the CL variable that receives the current local network ID. Specify the name of the character variable with a minimum length of 8 characters. If the network ID has fewer characters than the variable allows, the value is padded on the right with blanks.

LCLCPNAME

Specifies the name of the CL variable that receives the current local control point. Specify the name of the character variable with a minimum length of 8 characters. If the control point has fewer characters than the variable allows, the value is padded on the right with blanks.

LCLLOCNAME

Specifies the local location name.

DFTMODE

Specifies the name of the CL variable that receives the default mode name. Specify the name of the character variable with a minimum length of 8 characters. If the default mode has fewer characters than the variable allows, the value is padded on the right with blanks.

NODETYPE

Specifies the name of the CL variable that receives the current APPN nodetype. Specify the name of the character variable with a minimum length of 8 characters.

The values that can be returned in the variable as the node type are:

***ENDNODE**

The node does not provide network services to other nodes but may participate in the APPN network by using the services of an attached network server or may operate in a peer-to-peer environment similar to migration end nodes.

***NETNODE**

The node provides intermediate routing, route selection services, and distributed directory services for local users and to end nodes and migration end nodes that it is serving.

***BEXNODE**

The node performs as a branch extender node. It performs as an end node in the backbone APPN network, and performs as a network node server to end nodes within its local domain.

DTACPR

Specifies the name of the CL variable that receives the current level of data compression. Specify the name of the decimal variable with a minimum length of 10 digits without decimal positions.

The values that can be returned in the variable as the data compression level are:

- 0** *NONE: Data compression is not allowed on the session.
- 1** *REQUEST: Data compression is requested on the session by the local system. However, the request can be refused or changed to a lower compression level by the remote system. Data compression is allowed on the session if requested by the remote system.
- 2** *ALLOW: Data compression is allowed on the session by the local system if requested by a remote system. The local system does not request compression.
- 3** *REQUIRE: Data compression is required on the session. If the remote system does not change the levels of compression to the local system's exact requested levels, the session is not established. The data compression levels that the local system requires are the specified levels.

DTACPRINM

Specifies the name of the CL variable that receives the current level of intermediate node data compression. Specify the name of the decimal variable with a minimum length of 10 digits without decimal positions.

The values that can be returned in the variable as the intermediate node data compression levels are:

- 0** *NONE: The remote systems are not notified of a need to compress data when the iSeries 400 is an SNA intermediate node.
- 1** *REQUEST: The remote systems are requested to compress data when the iSeries 400 is an SNA intermediate node.

MAXINTSSN

Specifies the name of the CL variable that receives the maximum number of intermediate sessions. Specify the name of the decimal variable with a minimum length of 5 digits without decimal positions.

RAR Specifies the name of the CL variable that receives the route addition resistance. Specify the name of the decimal variable with a minimum length of 5 digits without decimal positions.

NETSERVER

Specifies the name of the CL variable that receives the list of network node servers. Specify the name of a character variable with a minimum length of 85 characters. If the server control point name or network ID has fewer characters than the variable allows, the value is padded on the right with blanks. The list can contain five node servers. Each server has the form: Network ID (9 characters) followed by the server name (8 characters). There are no separators. The network ID may contain the value *LCLNETID which specifies that the current network ID is used. If less than five node servers are specified, the remaining ones contain hexadecimal zeros for a name. As soon as the first null name is encountered in the list, it is safe to assume that the remaining names will also be null.

ALRSTS

Specifies the name of the CL variable that receives the current system alert status. Specify the name of a character variable with a minimum length of 10 characters. If the alert status value has fewer characters than the variable allows, the value is padded on the right with blanks.

The values that can be returned in the variable as the current system alert status are:

***ON** Alert processing is currently active.

***UNATTEND**

Alert processing is currently active, but the system is unattended.

***OFF** Alert processing is currently inactive.

ALRPRIFP

Specifies the name of the CL variable that receives the primary alert focal point. Specify the name of a character variable with a minimum length of 10 characters. If the primary focal point value has fewer characters than the variable allows, the value is padded on the right with blanks.

The values that can be returned in the variable as the primary alert focal point are:

***NO** The system is not an alert primary focal point.

***YES** The system is an alert primary focal point.

ALRDFTFP

Specifies the name of the CL variable that receives the default alert focal point. Specify the name of a character variable with a minimum length of 10 characters. If the default focal point value has fewer characters than the variable allows, the value is padded on the right with blanks.

The values that can be returned in the variable as the default alert focal point are:

***NO** The system is not an alert default focal point.

***YES** The system is an alert default focal point.

ALRBCKFP

Specifies the name of the CL variable that receives the name of the system that provides alert focal point services if the primary focal point is unavailable. You must specify the name of a character variable with a minimum length of 10 characters. If the back up system name has fewer characters than the variable allows, the value is padded on the right with blanks.

ALRRQSFP

Specifies the name of the CL variable that receives the name of the system that is requested to provide alert focal point services. You must specify the name of a character variable with a minimum length of 10 characters. If the requesting system name has fewer characters than the variable allows, the value is padded on the right with blanks.

ALRFTR

Specifies the name of the CL variable that receives the name of the active alert filter. You must specify the name of a character variable with a minimum length of 10 characters. If the alert filter name has fewer characters than the variable allows, the value is padded on the right with blanks.

ALRFTRLIB

Specifies the name of the CL variable that receives the name of the library that contains the alert filter definition. Specify the name of a character variable with a minimum length of 10 characters. If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.

ALRLOGSTS

Specifies the name of the CL variable that receives the current alert logging status. Specify the name of a character variable with a minimum length of 10 characters.

The values that can be returned in the variable as the current alert logging status are:

***NONE**

Do not log alerts.

***LOCAL**

Log only locally generated alerts.

***RCV** Log only alerts received from other nodes.

***ALL** Log both locally generated and incoming alerts.

ALRCTL

Specifies the name of the CL variable that receives the name of the controller through which alert messages are sent to another system when alert processing is active. Specify the name of a character variable with a minimum length of 10 characters. If the alert controller name has fewer characters than the variable allows, the value is padded on the right with blanks.

The values that can be returned in the variable are:

***NONE**

There is no controller for alerts.

controller-name

Specifies the name of the controller being used for alerts in an alert controller session. This controller is ignored if the system has a primary or default alert focal point (if, for example, the node is in another system's sphere of control).

ALRHLCNT

Specifies the name of the CL variable that receives the maximum number of alerts that are created before the alerts are sent over the alert controller session (ALRCTL network attribute). The alerts are held (queued) by the system until the specified number of alerts have been created. This parameter can be used to manage alerts that are sent over a limited resource by reducing the number of times alerts are sent. Specify the name of a decimal variable with a total length of 5 digits without decimal positions. The maximum number of alerts that can be created before the alerts are sent is 32,767.

Note:

The ALRHLCNT network attribute only applies when the ALRCTL network attribute is used. When management services sessions, APPN, and sphere of control support are used, the ALRHLCNT value is ignored.

The value that can be returned in the variable as the maximum number of alerts is:

***NOMAX**

The alerts are held indefinitely. The current alert hold count is the maximum value. The alerts can be sent at a later time by changing the ALRHLCNT value to a lower value.

MSGQ

Specifies the name of the CL variable that receives the group message queue name. This must be a character variable with a minimum length of 10 characters. If the message queue name has fewer characters than the variable allows, the value is padded on the right with blanks.

MSGQLIB

Specifies the name of the CL variable that receives the name of the library that contains the system-default network message queue. Specify the name of a character variable with a minimum length of 10 characters. If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.

OUTQ Specifies the name of the CL variable that receives the system default network output queue name. Specify the name of a character variable with a minimum length of 10 characters. If the output queue name has fewer characters than the variable allows, the value is padded on the right with blanks.

OUTQLIB

Specifies the name of the CL variable that receives the name of the library that contains the system-default network message queue. Specify the name of a character variable with a minimum length of 10 characters. If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.

JOBACN

Specifies the name of the CL variable that receives the current job action for input streams received through the network. Specify the name of a character variable with a minimum length of 10 characters. If the job action value has fewer characters than the variable allows, the value is padded on the right with blanks.

The values that can be returned in the variable as the current job action are:

***FILE** Input streams received from the network are filed in the queue of network files for the user to which it was sent.

***REJECT**
Input streams received from the network are rejected.

***SEARCH**
The table of network job entries is searched to determine the action for any input streams received.

MAXHOP

Specifies the name of the CL variable that receives the maximum number of systems in the SNADS network so that a distribution queue originating at this node may be received and rerouted on the path to its final destination. Specify the name of a decimal variable with a total length of 5 digits without decimal positions.

DDMACC

Specifies the name of the CL variable that receives the current system action for DDM or DRDA requests from other systems. Specify the name of a character variable with a minimum length of 10 characters. If the DDM access value has fewer characters than the variable allows, the value is padded on the right with blanks.

The values that can be returned in the variable as the current system action for DDM requests are:

***REJECT**
This system does not allow DDM or DRDA requests from remote systems. However, this system can still use DDM or DRDA to access files or SQL tables on remote systems. A remote system cannot access files or SQL tables on a system that specifies *REJECT.

***OBJAUT**
If the user profile associated with the DDM or DRDA job is authorized to the files, all file or remote SQL requests are accepted. Object authorities such as read, write, or update must also exist for the files.

program-name

Specifies the name of the customer validation program that can supplement object level security. This user-exit program can restrict user access to *PUBLIC and private files. The target DDM support calls the user program for each reference to a file. The user-exit program indicates to DDM if the request should proceed or end. iSeries 400 object level security still applies.

DDMACCLIB

Specifies the name of the CL variable that receives the name of the library that contains the DDM access program. Specify the name of a character variable with a minimum length of 10 characters. If the library name has fewer characters than the variable allows, the value is padded on the right with blanks. If *REJECT or *OBJAUT is returned for the DDMACC parameter, the value for DDMACCLIB is all blanks.

PCSACC

Specifies the name of the CL variable that receives the current system action for PCS requests. Specify the name of a character variable with a minimum length of 10 characters. If the PCS access value has fewer characters than the variable allows, the value is padded on the right with blanks.

The values that can be returned in the variable as the current system action for PCS requests are:

***REJECT**

The system does not allow any PCS requests.

***OBJAUT**

All PCS requests are allowed and controlled by the object authorizations on the system.

***REGFAC**

The registration facility is used to determine exit programs for the different servers. If no program is defined in the registration facility, *OBJAUT is used.

program-name

Specify the name of the customer supplied Client Access/400 host system application exit program that can supplement system object level security.

PCSACCLIB

Specifies the name of the CL variable that receives the name of the library that contains the PCS access program. Specify the name of a character variable with a minimum length of 10 characters. If the library name has fewer characters than the variable allows, the value is padded on the right with blanks. If *REJECT or *OBJAUT is returned for the PCSACC parameter, the value for PCSACCLIB is all blanks.

DFTNETTYPE

Specifies the name of the CL variable that receives the system default value for the integrated services digital network (ISDN) network type. The user must specify a character variable name with a minimum length of 10 characters.

The values that can be returned in the variable as the network type are:

***ATT** This value is used when attaching to an ISDN that uses an AT&T interface.

***DBP1TR6** This value is used when attaching to an ISDN controlled by Germany's Post Telephone and Telegraph Administration (PPT). (Deutsche Bundespost 1TR6).

***ETSI** This value is used when attaching to the ISDN that uses a European interface (ETSI, also known as EuroISDN standard).

***JAPAN** This value is used when attaching to an ISDN that uses a Japanese interface.

***NISDN** This value is used when attaching to an ISDN that uses a North American interface.

***NORTEL** This value is used when attaching to an ISDN that uses the Northern Telecom interface.

***NORTHAMT1** This value is used when attaching to a channelized North American T1 interface.

DFTCNNLST

Specifies the name of the CL variable that receives the system default value for the ISDN connection list. The user must specify a character variable name with a minimum length of 10 characters.

ALWANYNET

Specifies the name of the CL variable that receives the network attribute that allows iSeries 400 Communications APIs to use other communication transports that are not native for that API. Examples include ICF over TCP/IP or Sockets over SNA. You must specify a character variable name with a maximum length of 10 characters.

NWSDOMAIN

Specifies the name of the CL variable that receives the LAN Server domain to which all Integrated PC Servers, also known as File Server Input/Output Processors (FSIOP), on the system belong. You must specify a character variable name with a maximum length of 8 characters.

ALWVRTAPPN

Specifies the name of the CL variable that receives the current setting for the virtual APPN support. The character variable must have a minimum length of 10 characters.

ALWHPRTWR

Specifies the name of the CL variable that receives the current setting for the HPR tower transport support. The character variable must have a minimum length of 10 characters.

VRTAUTODEV

Specifies the name of the CL variable that receives the current setting for the maximum amount of automatically created APPC devices allowed on a virtual controller. The decimal variable must be of at least 5 digits length.

HPRPTHMR

Specifies the name of the CL variable that receives the current settings for the maximum amount of time in minutes for the HPR path switch timers. This field requires a 40 character variable, each 10 characters represents one of the four timer values in the order of network, high, medium and low priority.

ALWADDCLU

Specifies whether this system will allow another system to add it as a node in a cluster. The character variable must have a minimum length of 10 characters.

The values that can be returned in the variable for allow add to cluster are:

***NONE**

No other system can add this system as a node in a cluster.

***ANY** Any other system can add this system as a node in a cluster.

***RQSAUT**

Any other system can add this system as a node in a cluster only after the cluster add request has been authenticated.

MDMCNTRYID

Specifies the country or region identifier associated with a modem. The character variable must have a minimum length of 2 characters.

This defines the country-specific or region-specific default characteristics for modems which are internal to iSeries 400 I/O adapters. This value must be configured correctly to insure proper

operation and, in some countries or regions, meet legal requirements. The adapter will fail the vary on of the line if modem country or region ID is not set.

Examples for RTVNETA

Example 1: Retrieving Current System Name and Number of Systems

```
DCL VAR(&SNAME) TYPE(*CHAR) LEN(8)
DCL VAR(&HOPS) TYPE(*DEC) LEN(5 0)
RTVNETA SYSNAME(&SNAME) MAXHOP(&HOPS)
```

This command retrieves the current system name and the maximum number of systems in a SNADS network so that a distribution queue entry can be received and rerouted on the path to its destination.

Example 2: Retrieving Virtual APPN Support, APPC Device Limits, and HPR Path Switch Timers

```
DCL VAR(&ALWVRTAPPN) TYPE(*CHAR) LEN(10)
DCL VAR(&VRTAUTODEV) TYPE(*DEC) LEN(5 0)
DCL VAR(&HPRPHTMR) TYPE(*CHAR) LEN(40)
RTVNETA ALWVRTAPPN(&ALWVRTAPPN) VRTAUTODEV(&VRTAUTODEV) +
        HPRPHTMR(&HPRPHTMR)
```

This command retrieves the current network attribute settings for allow virtual APPN support, automatically created APPC devices on a virtual controller, and the HPR path switch timers.

Error messages for RTVNETA

*ESCAPE Messages

CPF1844

Cannot access network attribute &1.

RTVOBJD (Retrieve Object Description) Command Description

RTVOBJD Command syntax diagram

Purpose

The Retrieve Object Description (RTVOBJD) command is used in a CL program or REXX procedure to retrieve the description of a specific object.

» Restrictions:

1. The user must have EXECUTE authority to the library and some authority (that is, any one of the authorities other than *EXCLUDE) to the object. If the object is a file, you must have *OBJOPR authority to the object.
2. The user must have *EXECUTE authority for the auxiliary storage pool (ASP) device if a specific ASP device name or *ALLAVL is specified on the ASPDEV parameter. <<

Required Parameters

OBJ Specifies the qualified name of the object being retrieved. If no library name is given, *LIBL is used to find the specified object.

The name of the object being retrieved can be qualified by one of the following library values:

***LIBL:** » All libraries in the thread's library list are searched until the first match is found. If ASPDEV is specified when this value is used, ASPDEV(*) is the only valid value. <<

***CURLIB:** >> The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched. If ASPDEV is specified when this value is used, ASPDEV(*) is the only valid value. <<

library-name: Specify the name of the library to be searched.

object-name: Specify the name of the object that is being retrieved.

OBJTYPE

Specify the type of object being retrieved. If a variable is specified, it can be up to 8 characters in length. More information on this parameter is in Commonly used parameters.

Optional Parameters

>> ASPDEV

Specifies the auxiliary storage pool (ASP) device name where storage is allocated for the library of the object being retrieved (OBJ parameter). If the library resides in an ASP that is not part of the thread's library name space, this parameter must be specified to ensure the correct library is searched. ASPDEV(*) is the only valid value if *CURLIB or *LIBL is specified as the library qualifier on the OBJ parameter.

Single Values

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

***ALLAVL:** All available ASPs will be searched to locate the library. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and all available primary and secondary ASPs (ASPs 33-255 with a status of 'Available'). The ASP groups are searched in alphabetical order by the primary ASP. The system ASP and all defined basic user ASPs are searched after the ASP groups. ASPs and libraries to which the user is not authorized are bypassed and no authority error messages are sent. The search ends when the first object is found of the specified object name, library name and object type. If the user is not authorized to the object, an authority error message is sent.

***CURASPGRP:** If the thread has an ASP group, the primary and secondary ASPs in the ASP group will be searched to locate the library. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched.

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

Element 1: Auxiliary storage pool (ASP) device

auxiliary-storage-pool-device-name: The device name of the primary or secondary ASP to be searched. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Available'. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched.

Element 2: Auxiliary storage pool (ASP) search type

***ASP:** Specifies that only the single auxiliary storage pool (ASP) device named in element 1 is to be searched

***ASPGRP:** Specifies that the entire group of the primary auxiliary storage pool (ASP) device named in element 1 is to be searched. <<

RTNLIB

Specifies the name of a variable used to return the name of the library that contains the object. In a CL program, the variable returned has a length of 10 characters. If *LIBL or *CURLIB is specified for the library name on the OBJ parameter, the value returned is the name of the library where the object was found.

OBJATR

Specifies the name of a variable used to return an extended attribute of the object, such as a program or file type. In a CL program the variable returned has a length of 10 characters. For example, the variable may be returned with PROD (production) or CLP (control language program). No asterisk (*) precedes the value.

USRDFNATR

Specifies the name of a variable used to return the user-defined attribute of an object. In a CL program the variable returned has a length of 10 characters. Blanks are returned if the retrieved object does not have a user-defined attribute.

TEXT Specifies the text that briefly describes the object being retrieved. More information on this parameter is in Commonly used parameters.

OWNER

Specifies the name of a variable that is used to return the name of the object owner's user profile. In a CL program, the variable returned has a length of 10 characters.

PGP Specifies the name of a variable that is used to return the name of the primary group for the object. If there is no primary group for the object, this field contains a value of *NONE. In a CL program, the variable returned has a length of 10 characters.

ASP Specifies the name of a variable that is used to retrieve the number of the auxiliary storage pool (ASP) where storage is allocated for the object. In a CL program, the variable returned has a decimal length of (2 0). >> This variable will contain an ASP number up to 32. If an ASP number is larger than 32, -1 is returned in this variable. Parameter OBJASPDEV should be used to return the ASP device name when the ASP number is larger than 32. << The following values can be returned:

- 1 The object is in the system auxiliary storage pool.
- 2-32 The object is in a >> basic << user auxiliary storage pool.
- >> -1 The object is in a primary or secondary auxiliary storage pool with an ASP number larger than 32. The ASP device name for the primary or secondary ASP can be retrieved with the OBJASPDEV parameter. <<

>> **LIBASP**

Specifies the name of a variable that is used to retrieve the number of the auxiliary storage pool (ASP) where storage is allocated for the library containing the object. In a CL program, the variable returned has a decimal length of (5 0). This variable will contain an ASP number up to 32. If an ASP number is larger than 32, -1 is returned in this variable. Parameter LIBASPDEV should be used to return the ASP device name when the ASP number is larger than 32. The following values can be returned:

- 1 The library is in the system auxiliary storage pool.
- 2-32 The library is in a basic user auxiliary storage pool.
- 1 The library is in a primary or secondary auxiliary storage pool with an ASP number larger than 32. The ASP device name for the primary or secondary ASP can be retrieved with the LIBASPDEV parameter. <<

» OBJASPDEV

Specifies the name of a variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the object. In a CL program, the variable returned has a length of 10 characters. The following special values can be returned:

*N The name of the ASP device can not be determined.

*SYSBAS

The object is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32). <<

» LIBASPDEV

Specifies the name of a variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the library. In a CL program, the variable returned has a length of 10 characters. The following special values can be returned:

*N The name of the ASP device can not be determined.

*SYSBAS

The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32). <<

OVFASP

Specifies the name of a variable that is used to return the Object Overflowed ASP flag. In a CL program, the variable returned has a length of 1 character. A character value of '1' is returned if the object overflowed the ASP in which it resides. A character value of '0' is returned if the object does not overflow the ASP. It is not possible for an object residing in the system ASP » (ASP 1) or in a primary or secondary ASP (ASPs 33-255) << to overflow its ASP, therefore, a value of character value of '0' is always returned for objects in the system ASP » (ASP 1) or in a primary or secondary ASP (ASPs 33-255). <<

CRTDATE

Specifies the name of a variable used to return the date and time the object was created. In a CL program, the variable returned has a length of 13 characters. The value is returned in the form CYYMMDDHHMMSS, where C = century; '0' indicates years 19xx and '1' indicates years 20xx; YY = year, MM = month, DD = day, HH = hour, MM = minutes, and SS = seconds.

CHGDATE

Specifies the name of a variable used to return the date and time the object was last changed. In a CL program, the variable returned has a length of 13 characters. The variable is returned in the same format as CRTDATE or is returned blank if the object has not been changed.

SAVDATE

Specifies the name of a variable used to return the date and time the object was last saved. In a CL program, the variable returned has a length of 13 characters. The variable is returned in the same format as CRTDATE or is returned blank if the object has not been saved.

SAVACTDATE

Specifies the name of a variable used to return the date and time the object was last saved. In a CL program, the variable returned has a length of 13 characters. Blanks are returned if the object has not been saved or if SAVACT(*NO) was specified on the last save operation for the object.

RSTDATE

Specifies the name of a variable used to return the date and time the object was last restored. In a CL program, the variable returned has a length of 13 characters. The variable is returned in the same format as CRTDATE or is returned blank if the object has not been restored.

CRTUSER

Specifies the name of a variable used to return the name of the user that created the object. In a CL program, the variable returned has a length of 10 characters.

CRTSYSTEM

Specifies the name of a variable used to return the name of the system on which the object was created. In a CL program, the variable returned has a length of 8 characters.

OBJDMN

Specifies the name of a CL variable used to return the object domain value for an object. In a CL program, the variable returned has a length of 2 characters. A value of *U indicates the object is in the user domain; a value of *S indicates the object is in the system domain.

USEUPD

Specifies the name of a variable that indicates whether the object usage information is updated for this object type. In a CL program, the variable returned has a length of 1 character. The indicator is returned as 'Y' or 'N'. If 'N' is returned, the last used date for the object is blank.

USEDATE

Specifies the name of a 7-character CL variable used to return the date the object was used last. In a CL program, the variable returned has a length of 7 characters. The date is returned in the form CYYMMDD. If the object does not have a last used date, blanks are returned.

USECOUNT

Specifies the name of a variable used to return the number of days the object has been used. In a CL program, the variable returned has a decimal length of (5 0). If the object does not have a last used date, zero (0) is returned.

RESETDATE

Specifies the name of a variable used to return the date on which the days used count was reset to 0. In a CL program, the variable returned has a length of 7 characters. The date is returned in the form CYYMMDD. If the days used count has not been reset, blanks are returned.

STG

Specifies the name of a variable used to return the storage status of the object data. In a CL program, the variable returned has a length of 10 characters. *FREE is returned if the object data has been freed and the object has been suspended. *KEEP is returned if the object data has not been freed and the object has not been suspended.

CPR

Specifies the name of a variable used to return the compression status of the object. In a CL program, the variable returned has a length of 1 character. A character value of 'Y' is returned if the object is compressed. A character value of 'N' is returned if the object is permanently decompressed. A character value of 'T' is returned if the object is temporarily decompressed. A character value of 'F' is returned if the object is eligible for compression but is saved with storage freed. A character value of 'X' is returned if the object is ineligible for compression.

SIZE

Specifies the name of a variable used to return the size of the object in bytes. In a CL program, the variable returned has a decimal length of (15 0).

SAVSIZE

Specifies the name of a variable used to return the size of the object in bytes of storage at the time of the last save operation. In a CL program, the variable returned has a decimal length of (15 0). The variable is returned with zeros if the object has not been saved.

SAVCMD

Specifies the name of a variable used to return the command used to save the object. In a CL program, the variable returned has a length of 10 characters. The variable is returned blank if the object has not been saved.

SAVSEQNBR

Specifies the name of a variable used to return the tape sequence number generated when the object was saved on tape. In a CL program, the variable returned has a decimal length of (4 0). The variable is returned with zeros if the object has not been saved or was not saved to tape. This variable will contain a sequence number up to 9999. If a sequence number is actually greater than 9999, -5 is returned in this variable. Parameter SAVLRGSEQ should be used to return a sequence number which can be larger than 9999.

SAVLRGSEQ

Specifies the name of a variable used to return the tape sequence number (similar to parameter SAVSEQNBR). This variable can contain a larger tape sequence number than parameter SAVSEQNBR.

SAVVOL

Specifies the name of a variable used to return the tape, diskette, or optical volumes used for saving the object. In a CL program, the variable returned has a length of 71 characters. The variable returns up to 10 six-character volumes. The volume IDs begin in character positions 1, 8, 15, 22, 29, 36, 43, 50, 57, and 64. Each volume ID entry is separated by a single character. If the object was saved in parallel format, the separator character contains a '2' before the first volume in the second media file, a '3' before the third media file, and so on, up to a '0' before the tenth media file. Otherwise, the separator characters are blank. If more than 10 volumes are used and the object was saved in serial format, '1' is returned in the 71st character of the variable. If the object was saved in parallel format, a '2' is returned in the 71st character of the variable. Otherwise, the 71st character is blank. The variable is returned blank if the object was last saved to a save file, or if it was never saved.

SAVDEV

Specifies the name of a variable used to return the type of device to which the object was last saved. In a CL program, the variable returned has a length of 10 characters. The variable is returned with one of the following values, dependent on the device used for the last save operation:

- *SAVF for a save file.
- *DKT for a diskette.
- *TAP for a tape.
- *OPT for an optical volume.
- The variable is returned blank if the object was not saved.

SAVF Specifies the name of a variable used to return the name of the save file if the object was saved to a save file. In a CL program, the variable returned has a length of 10 characters. The variable is returned blank if the object was not saved to a save file.

SAVFLIB

Specifies the name of a variable used to return the name of the library that contains the save file if the object was saved to a save file. In a CL program, the variable returned has a length of 10 characters. The variable is returned blank if the object was not saved to a save file.

SAVLABEL

Specifies the name of a variable used to return the file label used when the object was saved. In a CL program, the variable returned has a length of 17 characters. The variable is returned blank if the object was not saved to tape, to diskette, or to an optical volume. The value of the return variable corresponds to the value specified on the LABEL parameter on the command used to save the object.

SRCF Specifies the name of a variable used to return the name of the source file used to create the object. In a CL program, the variable returned has a length of 10 characters. The variable is returned blank if no source file was used to create the object.

SRCFLIB

Specifies the name of a variable used to return the name of the library that contains the source file that was used to create the object. In a CL program, the variable returned has a length of 10 characters. The variable is returned blank if no source file was used to create the object.

SRCMBR

Specifies the name of a variable used to return the name of the member in the source file (parameter SRCF). In a CL program, the variable returned has a length of 10 characters. The variable is returned blank if no source file was used to create the object.

SRCDATE

Specifies the name of a variable used to return the date and time the member in the source file was last updated. In a CL program, the variable returned has a length of 13 characters. The variable is returned in the same format as CRTDATE or is returned blank if the member is not updated.

SYSLVL

Specifies the name of a variable used to return the level of the operating system when the object was created. In a CL program, the variable returned has a length of 9 characters:

- A 3-character version level starting in character position 1
- A 3-character release level starting in character position 4
- A 3-character modification level starting in character position 7

The first character of the version level is always the letter 'V'; the first character of the release level is always the letter 'R'; the first character of the modification level is always the letter 'M'.

COMPILER

Specifies the name of a variable used to return the licensed program identifier, version level, release level, and modification level of the compiler. In a CL program, the variable returned has a length of 16 characters:

- The 7-character name starting in character position 1
- The 3-character version level in character position 8
- The 3-character release level in character position 11
- The 3-character modification level in character position 14

The first character of the version level is always the letter 'V'; the first character of the release level is always the letter 'R'; the first character of the modification level is always the letter 'M'. The variable is returned blank if no compiler was used.

OBJLVL

Specifies the name of a variable used to return the object control level for the created object. In a CL program, the variable returned has a length of 8 characters.

ALWAPICHG

Specifies the name of a variable used to return the Allow Change by Program flag. A '1' is returned if the object can be changed with the Change Object Description API, QLICOBJD. A '0' is returned if the object cannot be changed with the API.

APICHG

Specifies the name of a variable used to return the Changed by Program flag. A '1' is returned if the object has been modified with the Change Object Description API, QLICOBJD. A '0' is returned if the object has not been modified by the API.

USRCHG

Specifies the name of a variable used to return whether the program was modified by the user. In a CL program, the variable returned has a length of 1 character. A '1' is returned if the object was modified by the user. A '0' is returned if the object was not modified by the user.

LICPGM

Specifies the name of a variable used to return the name, version level, release level, and modification level of the licensed program if the retrieved object is part of a licensed program. In a CL program, the variable returned has a length of 16 characters:

- The 7-character name starting in character position 1
- The 3-character version level in character position 8
- The 3-character release level in character position 11
- The 3-character modification level in character position 14

The first character of the version level is always the letter 'V'; the first character of the release level is always the letter 'R'; the first character of the modification level is always the letter 'M'. The variable is returned blank if the retrieved object is not a licensed program.

PTF Specifies the name of a variable used to return the Program Temporary Fix number that resulted in the creation of the retrieved object. In a CL program, the variable returned has a length of 10 characters. The variable is returned blank for user created-objects.

APAR Specifies the name of a variable used to return the Authorized Program Analysis Report identification. In a CL program, the variable returned has a length of 10 characters. The variable is returned with the APAR ID that caused this object to be patched. The variable is returned blank if the object has not been changed as a result of an APAR.


This field is not updated when IBM-supplied Program Temporary Fixes are applied. The field is changed in the following situations:


- The default for a command is changed with the Change Command Default (CHGCMDDFT) CL command. The field is set to CHGDFT.
- The Change Object Description API, QLICOBJD, can change this field to any value.

OBJAUD

Specifies the name of a 10-character variable used to return the auditing value of the object. The values that can be retrieved include *NONE, *USRPRF, *CHANGE, and *ALL. See the OBJAUD parameter on the Change Object Audit (CHGOBJAUD) command for more information.


OBJSIG

Specifies the name of a variable used to return whether the object  has an OS/400 digital signature. In a CL program, the variable returned has a length of 1 character. The following values can be returned:

- '1' - The object has an OS/400 digital signature.
- '0' - The object does not have an OS/400 digital signature. 


SYSSIG

Specifies the name of a variable used to return whether the object is signed by a source that is trusted by the system. The following values can be returned:

- '1' - The object is signed by a source that is trusted by the system. If the object has multiple signatures, at least one of the signatures came from a source that is trusted by the system.
- '0' - None of the object signatures came from a source that is trusted by the system. 

MLTSIG

Specifies the name of a variable used to return whether the object has more than one OS/400 digital signature. The following values can be returned:

- '1' - The object has more than one OS/400 digital signature. If the SYSSIG variable has a value of '1', at least one of the signatures is from a source trusted by the system.
- '0' - The object has only one OS/400 digital signature, or does not have an OS/400 digital signature. Refer to the OBJSIG variable to determine whether the object has an OS/400 digital signature. 

JRNSTS

Specifies the name of a variable used to return the current journaling status of the object. In a CL program, the variable returned has a length of 1 character. A character value of '1' is returned if the object is currently being journaled. A character value of '0' is returned if the object is currently not being journaled. In a CL program, the variable returned has a length of 1 character.

Note:

Other journal fields may contain data even though the object is not currently being journaled.

JRN Specifies the name of a variable used to return the name of the current or last journal. In a CL program, the variable returned has a length of 10 characters. The variable is returned blank if the object has never been journaled.

JRNLIB

Specifies the name of a variable used to return the name of the library that contains the journal. In a CL program, the variable returned has a length of 10 characters. The variable is returned blank if the object has never been journaled.

JRNIMG

Specifies the name of a variable used to return the journal image information. In a CL program, the variable returned has a length of 1 character. The following values can be returned dependent upon what images were generated for changes to the object:

- '0' - If only *after* images are written to the journal for changes to the object.
- '1' - If both *before* and *after* images are written to the journal for changes to the object.

This variable is returned blank if the object has never been journaled.

JRNOMTE

Specifies the name of a variable used to return information regarding journal entries to be omitted. In a CL program, the variable returned has a length of 1 character. The following values can be returned for the journal entries to be omitted.

- '1' - If *open* and *close* operations on the specified objects do not generate *open* and *close* journal entries.
- '0' - If no journal entries are omitted.

This variable is returned blank if the object has never been journaled.

JRNSTRDATE

Specifies the name of a variable used to return the date and time journaling last started. In a CL program, the variable returned has a length of 13 characters. The value is returned in the same format as CRTDATE or is returned blank if the object has never been journaled.

Example for RTVOBJD

Jane Brown enters the following command to create a library:

```
CRTLIB LIB(PGMLIB) TYPE(*PROD)
      TEXT('Library for test programs')
```

Later, Jane, or anyone else with the proper authority, can retrieve the attributes of the library in a CL program as shown below:

```
DCL &CRTDATE CHAR 13
DCL &OWN      CHAR 10
DCL &STORE   CHAR 10
      .
      .
      .
RTVOBJD OBJ(*LIBL/PGMLIB) OBJTYPE(*LIB) TEXT(&TEXT)
      CRTDATE(&CRTDATE) OWNER(&OWN) STG(&STORE)
```

The values returned in the variables of the CL program are shown below:

```
&TEXT      = Library for test programs
&CRTDATE   = 0900211130000
&OWN       = JBROWN
&STORE     = *KEEP
```

The value returned in the variable &CRTDATE indicates that PGMLIB was created on the eleventh day of February, 1990, at 1300 hours. The value returned in the variable &OWN indicates that the library was

created by user profile JBROWN. The value returned in the variable &STORE indicates that the data in PGMLIB has not been freed, and the library has not been suspended.

Error messages for RTVOBJD

***ESCAPE Messages**

» CPF8ED

Device description &1 not correct for operation. «

CPF2115

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

CPF2150

Object information function failed.

CPF2151

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

» CPF2173

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

» CPF218C

&1 not a primary or secondary ASP. «

» CPF218D

&1 not a primary ASP when *ASPGRP specified. «

CPF2451

Message queue &1 is allocated to another job.

CPF3202

File &1 in library &2 in use.

CPF3203

Cannot allocate object for file &1 in &2.

CPF36F7

Message queue QSYSOPR is allocated to another job.

» CPF980B

Object &1 in library &2 not available. «

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9807

One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

CPF9810

Library &1 not found.

CPF9811

Program &1 in library &2 not found.

CPF9812

File &1 in library &2 not found.

CPF9814

Device &1 not found.

CPF9820

Not authorized to use library &1.

CPF9821

Not authorized to program &1 in library &2.

CPF9822

Not authorized to file &1 in library &2.

CPF9825

Not authorized to device &1.

CPF9830

Cannot assign library &1.

CPF9831

Cannot assign device &1.

» CPF9833

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

RTVPWRSCDE (Retrieve Power On/Off Schedule Entry) Command Description

RTVPWRSCDE Command syntax diagram

Purpose

The Retrieve Power On/Off Schedule Entry (RTVPWRSCDE) command is used in a CL program or REXX procedure to retrieve a power on/off schedule entry value. The value is returned (copied) to the specified CL variable.

Optional Parameters

DAY Specifies the day for which the user is retrieving a power on/off schedule entry.

***TODAY:** The current date's schedule entry is retrieved.

***SUN:** The default values for Sunday are retrieved.

***MON:** The default values for Monday are retrieved.

***TUE:** The default values for Tuesday are retrieved.

***WED:** The default values for Wednesday are retrieved.

***THU:** The default values for Thursday are retrieved.

***FRI:** The default values for Friday are retrieved.

***SAT:** The default values for Saturday are retrieved.

date: Specify the date for which a schedule entry is to be retrieved. The date must be specified in the format as defined by the job.

PWRONTIME

Specifies the name of the CL variable that receives the power on time. The variable named has a minimum length of 6 characters. The special value *NONE or the time in the format hhmmss, where hh = hours, mm = minutes, and ss = seconds, is returned.

PWROFFTIME

Specifies the name of the CL variable that receives the power off time. The variable named has a minimum length of 6 characters. The special value *NONE or the time in the format hhmmss, where hh = hours, mm = minutes, and ss = seconds, is returned. *NONE or the time in the format hhmm, where hh = hours and mm is minutes, is returned.

DAYDESC

Specifies the name of the CL variable that receives the specific date description. The variable named has a minimum length of 38 characters.

MSGITV

Specifies the name of the CL variable that receives the message interval value. The message interval is the number of minutes before the scheduled power off that a message is sent to all work stations warning users of the impending power off. The variable named has a minimum length of 2 characters.

Examples for RTVPWRSCDE**Example 1: Retrieving Today's Schedule Entry**

```
DCL VAR(&ONTIME) TYPE(*CHAR) LEN(6)
DCL VAR(&OFFTIME) TYPE(*CHAR) LEN(6)

RTVPWRSCDE DAY(*TODAY) PWRONTIME(&ONTIME)
           PWROFFTIME(&OFFTIME)
```

This command retrieves the power on and off times for today.

Example 2: Retrieving Tuesday's Schedule Entry

```
DCL VAR(&ONTIME) TYPE(*CHAR) LEN(6)
DCL VAR(&OFFTIME) TYPE(*CHAR) LEN(6)

RTVPWRSCDE DAY(*TUE) PWRONTIME(&ONTIME)
           PWROFFTIME(&OFFTIME)
```

This command retrieves the defaults for Tuesday's schedule.

Error messages for RTVPWRSCDE***ESCAPE Messages****CPF1E2B**

Power scheduler and cleanup options not found.

CPF1E23

Power schedule or cleanup options in use by another user.

CPF1E99

Unexpected error occurred.

CPF2105

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

CPF9808

Cannot allocate one or more libraries on library list.

RTVPDGPRF (Retrieve Print Descriptor Group Profile) Command Description

RTVPDGPRF Command syntax diagram

Purpose

The Retrieve Print Descriptor Group Profile (RTVPDGPRF) command is used in a CL program or REXX procedure to retrieve one or more of the print descriptor group profile values associated with a user profile. The values are returned in the specified CL variables for the desired user.

Restriction: The program must have *OBJMGT authority to use this command.

Required Parameter

USER Specifies the name of the user whose print descriptor group (PDG) profile is being retrieved.

***CURRENT:** The user profile that is currently running is used.

user-name: Specify the name of the user whose PDG profile is being retrieved.

Optional Parameters

RTNUSER

Specifies a 10-character CL variable used to retrieve the name of user profile for which information is requested.

PDG Specifies a 10-character CL variable used to retrieve the print descriptor group of the user profile for which information is requested.

PDGLIB

Specifies a 10-character CL variable used to retrieve the library of print descriptor group of the user profile for which information is requested.

PRTD Specifies a 256 character CL variable used to retrieve the print descriptor name from the user profile for which information is requested.

Example for RTVPDGPRF

Assume a user with *OBJMGT authority entered the following command:

```
CHGUSRPRF USER(JWONG) PDG(*LIBL/TAXFORMS) PRTD(FORM_C1)
```

Also assume the program with *OBJMGT authority contains the following commands and declarations:

```
DCL VAR(&USER) TYPE(*CHAR) LENGTH(10)
DCL VAR(&GROUP) TYPE(*CHAR) LENGTH(10)
DCL VAR(&LIBRARY) TYPE(*CHAR) LENGTH(10)
DCL VAR(&DESCRIPT) TYPE(*CHAR) LENGTH(256)
```

```
RTVPDGPRF USER(JWONG) RTNUSER(&USER) PDG(&GROUP)
PDGLIB(&LIBRARY) PRTD(&DESCRIPT)
```

When the above program is called, the following values are returned:

```
&USER      'JWONG_ _ _ _ _'
&GROUP     'TAXFORMS_ _ _ _ _'
&LIBRARY   'TAXLIB_ _ _ _ _'
&DESCRIPT  'FORM_C1_..._ _ _ _ _'
```

Note:

The underlines (_) above represent blanks. The value returned in variable &DESCRIPT is FORM_C1 followed by 249 blanks.

Error messages for RTVPDGPRF

*ESCAPE Messages

CPF2204

User profile &1 not found.

CPF2217

Not authorized to user profile &1.

CPF2247

Internal security object not available. Reason code &1.



RTVPRD (Retrieve Product) Command Description

Note: To use this command, you must have the 5722-SM1 (System Manager for iSeries) licensed program installed.

RTVPRD Command syntax diagram

Purpose

The Retrieve Product (RTVPRD) command provides the capability to retrieve a product, packaged with the OS/400 product packaging support, from a managed system.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The Managed System Services licensed program and the OS/400 licensed program cannot be specified.
3. A product can only be retrieved from a single managed system.

Note:

A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

Required Parameter

PRDID

Specifies the 7-character identifier of the product to be retrieved.

product-ID: Specify the 7-character product ID that is used in the activity.

Optional Parameters

RLS Specifies which version, release, and modification level of the product is used.

***ONLY:** Only one version, release, and modification level is used for the product option.

version-release-modification: Specify the release level in the format VxRxMy, where Vx is the version number, where Rx is the release number, and My is the modification number. Valid values for x are the numbers 0 through 9. Valid values for y are the numbers 0 through 9 and the letters A through Z.

OPTION

Specifies which of the optional parts of the product given in the PRDID parameter are used.

***BASE:** Only the base part of the product is used.

product-option-number: Specify the option number for the product load being saved. Valid values range from 1 through 99.

LODTYPE

Specifies the product load objects being used.

***ALL**: Code and language objects specified on the LODID parameter are used.

***CODE**: The program objects associated with this product load are used.

***LNG**: Specify that the objects associated with the NLV identified on the LODID parameter are used.

LODID

Specifies the load identifier used.

***ALL**: All languages for this product option are saved.

***CODE**: The code load is used.

product-load-ID: Specify when LODTYPE(*LNG) or LODTYPE(*ALL) is used, the load ID must be one of the valid IBM national language versions and must be specified in the form 29xx.

CPNAME

Specifies the APPN control point names of the managed systems on which this activity is to be performed.

***NETATR**: The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

network-identifier: Specify the APPN network identifier of the managed system on which the activity is to be performed.

control-point-name: Specify the APPN control point name of the managed system on which the activity is to be performed.

TGTRLS

Specifies the release of the operating system on which you intend to use the product.

***CURRENT**: The object is used on the release of the operating system currently running on your system. If V5R2M0 is running on your system, *CURRENT means that you intend to use the object on a system with V5R2M0 installed. The object can also be used on a system with any later release of the operating system installed.

***PRV**: The object is intended for a system that is at the previous release level compared to the local system.

Note:

Modification levels are not supported. To specify the previous release with a modification level other than 0, such as V4R1M4, specify the value for the release rather than the special value *PRV.

release-level: Specify the release level in the VxRxMx format. The object is used on a system with the specified release or with any later release of the operating system installed.

Valid values depend on the current version, release, and modification level and they change with each new release.

Examples for RTVPRD

Example 1: Retrieving the Program and Language Parts

```
RTVPRD PRDID(1ACCOUN) RLS(V5R2M0) OPTION(*BASE)
      LODTYPE(*ALL) LODID(*ALL) CPNAME((*NETATR SYS1))
```

This command retrieves the program and language parts for the base option of product 1ACCOUN with release V5R2M0 from iSeries server SYS1.

Example 2: Retrieving an Option

```
RTVPRD PRDID(1CHECKS) RLS(V5R2M0) OPTION(10)
      LODTYPE(*ALL) LODID(*ALL) CPNAME((*NETATR SYS2))
```

This command retrieves the option 10 of the 1CHECKS product, release V5R2M0, for both the program and the language parts from the iSeries server SYS2.

Error messages for RTVPRD

*ESCAPE Messages

None <>>

RTVPTF (Retrieve Program Temporary Fix) Command Description

Note: To use this command, you must have the 5722-SM1 (System Manager for iSeries) licensed program installed.

RTVPTF Command syntax diagram

Purpose

The Retrieve Program Temporary Fix (RTVPTF) command allows a service provider to retrieve remote PTFs and cover letters from a service requester's system.

Note: A message is automatically returned identifying the name of the change request. You can use the Work with Submitted Change Request (WRKSBMCRQ) command to monitor the status.

Restriction: The retrieve function of this command is only supported when the service requester has the Managed System Services licensed program installed.

Required Parameters

PTFID Specifies the PTF that is to be retrieved. A maximum of 50 PTF identifiers can be specified.

Element 1: PTF identifier

ptf-identifier: Specify a 7-character PTF identifier.

Element 2: Product identifier

***ONLY:** The PTF identifier is associated with only one product.

product-id: Specify the product of the PTF. The product must be specified when PTF identifiers are not unique across products.

Element 3: Release level of product

release(VxRxMx): Specify the release level of the product. The release level must be specified when PTF identifiers are not unique across products. The format is VxRxMx.

DESTSRVRQS

Specifies the service requester where the PTF should be retrieved. Only one service requester can be specified or selected.

Element 1: Network identifier

***SELECT:** Shows a list of service requesters. From the list, the user can select a service requester from where the PTF is to be retrieved. *SELECT is not valid when the command is used in a batch environment.

***NETATR:** The remote network identifier of the service requester is the same as the network identifier of this system.

remote-network-identifier: Specify the remote network identifier of the service requester.

Element 2: Control Point

remote-control-point: Specify the remote control point name of the service requester.

Optional Parameters

PTFPART

Specifies if PTFs or cover letters should be retrieved.

***ALL:** Retrieve the PTF and its cover letter in the language specified.

***PTF:** Retrieve only the PTF.

***CVRLTR:** Retrieve only the cover letter in the language specified.

CVRLTRLNG

Specifies the cover letter language that should be retrieved.

***SRVPVDA:** The value set in the Change Service Provider Attributes (CHGSRVPVDA) command is used.

cover-letter-language-ID: Specify the cover letter language code. Prompting (F4) on this field provides a list of all possible cover letter language codes.

Note:

If the specified cover letter language is not found on the service requester's system, a default cover letter language of 2924, which is uppercase and lowercase English, or 2950, which is uppercase English, is retrieved if available.

Examples for RTVPTF

Example 1: Retrieving a PTF and Cover Letter

```
RTVPTF PTFID(SF00001) DESTSRVRQS(*NETATR CHICAGO1)
```

This example shows how to retrieve a PTF and a cover letter from a service requester. The cover letter language to be retrieved is defined as a service provider attribute.

Example 2: Retrieving a Cover Letter

```
RTVPTF PTFID(123456) DESTSRVRQS(*SELECT)
PTFPART(*CVRLTR) CVRLTRLNG(2962)
```

This example shows how to retrieve a cover letter from a selected service requester in the 2962 language.

Error messages for RTVPTF

*ESCAPE Messages

RTVQMFORM (Retrieve Query Management Form) Command Description

RTVQMFORM Command syntax diagram

Purpose

The Retrieve Query Management Form (RTVQMFORM) command allows the user to retrieve encoded form source from a query management form (QMFORM) object. The source records are placed into a source file member that can be edited.

Form source can also be retrieved from a query definition (QRYDFN) object when the QMFORM specified by the user does not exist.

Required Parameters

QMFORM

Specifies the name of the query management form object whose source is being retrieved.

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

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

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

library-name: Specify the name of the library to be searched.

form-name: Specify the name of the form being retrieved.

SRCFILE

Specifies the qualified name of the previously created source physical file into which the encoded form source records are being written.

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

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

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

library-name: Specify the name of the library to be searched.

source-file-name: Specify the name of the source file that is to receive the form source.

Optional Parameters

SRCMBR

Specifies the name of the source physical file member into which the encoded form source records are being written. If a source file member name is not specified, then the form name specified on the QMFORM parameter is used.

If the member existed before running this command, it is cleared before any source statements are written into it. If the member does not exist, it is created.

***QMFORM:** The member name is the same as the form name specified on the QMFORM parameter.

source-file-member-name: Specify the name of the member to receive the form source.

ALWQRYDFN

Specifies whether form information is taken from a QRYDFN object when a query management form (QMFORM) object cannot be found using the object name specified by the user. Any information that has to be derived in this way is discarded when the command has completed processing. No query management object is created.

***NO:** Information is not taken from a QRYDFN object.

***YES:** Information is taken from a QRYDFN object when the specified QMQRY object is not found.

***ONLY:** Information is taken only from a QRYDFN object. Query management objects are ignored.

Examples for RTVQMFORM

Example 1: Retrieving Encoded Form Source

```
RTVQMFORM QMFORM(RPTLIB/SALFORM) SRCFILE(FORMS)
SRCMBR(EMPFORM)
```

This command retrieves the encoded form source from the form named SALFORM located in the RPTLIB library. The encoded form source records that are retrieved are placed into the newly created or cleared member EMPFORM in the first file named FORMS in the user's library list.

Example 2: Retrieving Source From Either the QMFORM or the QRYDFN

```
RTVQMFORM QMFORM(RPTLIB/SALFORM) SRCFILE(FORMS)
SRCMBR(EMPFORM) ALWQRYDFN(*YES)
```

This command retrieves the encoded form source from the query management form (QMFORM) named SALFORM located in the RPTLIB library. If there is no QMFORM object named SALFORM in the RPTLIB library, then the form source is retrieved from the query definition (QRYDFN) named SALFORM in the RPTLIB library. The encoded form source records that are retrieved are placed into the first file named FORMS in the user's library list.

Error messages for RTVQMFORM

*ESCAPE Messages

QWM2701

&1 command failed.

QWM2703

&1 command ended.

QWM2705

Source file &1 in &2 not available.

RTVQMORY (Retrieve Query Management Query) Command Description

RTVQMORY Command syntax diagram

Purpose

The Retrieve Query Management Query (RTVQMORY) command allows the user to retrieve Structured Query Language (SQL) source from a query management query (QMORY) object. The source records are placed into a source file member that can be edited.

The user can also retrieve query source from a query definition (QRYDFN) object when the QMORY specified by the user does not exist.

Required Parameters

QMORY

Specifies the name of the query management query whose source is being retrieved.

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

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

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

library-name: Specify the name of the library to be searched.

query-name: Specify the name of the query being retrieved.

SRCFILE

Specifies the qualified name of the previously created source physical file into which the query source records are being written.

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

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

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

library-name: Specify the name of the library to be searched.

source-file-name: Specify the name of the source file to receive the query source being retrieved.

Optional Parameters

SRCMBR

Specifies the name of the member into which the query source records are being written. If a source file member name is not specified, then the query name specified on the QMQRYPARAMETER is used.

If the member existed before running this command, it is cleared before any source records are written to it. If the member does not exist, it is created.

***QMQRYPARAMETER:** The member name is the same as the query name specified on the QMQRYPARAMETER parameter.

source-file-member-name: Specify the name of the member to receive the query source.

ALWQRYPDFN

Specifies whether query information is taken from a QRYDFN object when a QMQRYPARAMETER object cannot be found using the object name specified by the user. Any information that has to be derived in this way is discarded when the command has completed processing. No query management object is created.

***NO:** Information is not taken from a QRYDFN object.

***YES:** Information is taken from a QRYDFN object when the specified QMQRYPARAMETER object is not found.

***ONLY:** Information is taken only from a QRYDFN object. Query management objects are ignored.

Examples for RTVQMQRYPARAMETER

Example 1: Retrieving SQL Source

```
RTVQMQRYPARAMETER QMQRYPARAMETER(RPTLIB/SALQRYPARAMETER) SRCFILE(QRYS)
SRCMBR(EMPQRYPARAMETER)
```

This command retrieves the source from the query named SALQRYPARAMETER located in the RPTLIB library. The source records that are retrieved are placed into the newly created or cleared member EMPQRYPARAMETER in the first file named QRYS in the user's library.

Example 2: Retrieving Source From Either the QMQRYPARAMETER or the QRYDFN

```
RTVQMQRYPARAMETER QMQRYPARAMETER(RPTLIB/SALQRYPARAMETER)
SRCFILE(QRYS) SRCMBR(EMPQRYPARAMETER)
ALWQRYPDFN(*YES)
```

This command retrieves the source from the query management query (QMQRYPARAMETER) named SALQRYPARAMETER in the RPTLIB library. If there is no QMQRYPARAMETER object named SALQRYPARAMETER in the RPTLIB library, then the query source is retrieved from the query definition (QRYDFN) named SALQRYPARAMETER in the RPTLIB library. The source records are placed into the newly created or cleared member EMPQRYPARAMETER in the first file named QRYS in the user's library list.

Error messages for RTVQMQRYPARAMETER

*ESCAPE Messages

QWM2701

&1 command failed.

QWM2703

&1 command ended.

QWM2705

Source file &1 in &2 not available.

RTVSFWPKGA (Retrieve Software Package Attributes) Command Description

RTVSFWPKGA Command syntax diagram

Purpose

The Retrieve Software Package Attributes (RTVSFWPKGA) command retrieves information from an AIX Install package and builds an install table of contents (.toc) file in the designated directory.

Restriction: This command is shipped with public *EXCLUDE authority and the QSRV user profile has private authority to use the command.

Parameters

NWSD

Specifies the network server description (NWSD) to be used. The NWSD must be varied on and operational. The NWSD retrieves information from packages in the specified directory and builds an AIX Install Table Of Contents File in the specified directory.

network-server-description-name: Specify the network server description to be used to retrieve the software package attributes.

DIR Specifies the directory for which the install table of contents is built.

***AIXUSER:** The install table of contents is built for the system-supplied AIX user directory
/QOpenSys/QIBM/UserData/AIXIntegration/AllServers/InstallPackages

***AIXPRD:** The install table of contents is built for the system-supplied product directory
/QOpenSys/QIBM/Prod/AIXIntegration/Fixes

'directory-name': Specify the directory name for which the install table of contents is built.

Example for RTVSFWPKGA

```
RTVSFWPKGA  NWSD(AIXNWSD)
             DIR('/QOpenSys/myAIXsoftware')
```

This command builds the install table of contents using the network server description AIXNWSD for all software in the directory.

/QOpenSys/myAIXsoftware

RTVSWLSRC (Retrieve Stop Word List Source) Command Description

RTVSWLSRC Command syntax diagram

Purpose

The Retrieve Stop Word List Source (RTVSWLSRC) command is used to retrieve the words from an IBM-supplied or user-created stop word list into a source file.

Required Parameters

LANGID

Specifies the language identifier (ID) for the stop word list.

SRCFILE

Specifies the qualified name of the source file used to receive the stop word list words. The contents of the source file are replaced.

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

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

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

library-name: Specify the name of the library to be searched.

source-file-name: Specify the name of the source file.

Optional Parameters

SRCMBR

Specifies the name of the source file member used to receive the stop word list words. The member is in the source file specified on the SRCFILE parameter.

***LANGID:** The language ID is used as the source file member name.

source-file-member-name: Specify the name of the member in the source file used to receive the stop word list.

TYPE Specifies the type of stop word list being retrieved

***IBM:** The stop word list is IBM-supplied.

***USER:** The stop word list is user-created.

Example for RTVSWLSRC

```
RTVSWLSRC LANGID(ENG) SRCFILE(MYLIB/MYFILE)
```

This command retrieves the stop word list into source file MYFILE in library MYLIB that has the language ID ENG.

Error messages for RTVSWLSRC

*ESCAPE Messages

CPF8723

Record length too small for database source file.

CPF8725

&1 type stop word list not supported for language.

CPF9899

Error occurred during processing of command.

RTVSYISINF (Retrieve System Information) Command Description

RTVSYISINF Command syntax diagram

Purpose

The Retrieve System Information (RTVSYISINF) command retrieves the following information from the system and places the information in save files and user spaces in the library specified on the command.

1. Edit descriptions - QIZAEDTDSF *SAVF
2. Network attributes - QIZANETA *USRSPC
3. Reply list entries - QIZARLESP *USRSPC
4. Service attributes - QIZASRVASP *USRSPC (V3R1M0 and V3R2M0 systems only)
5. Service providers - QIZASRVASF *SAVF (V3R1M0 and V3R2M0 systems only)
6. System values - QIZASYSVAL *USRSPC

No error message will be displayed if any of the above information does not exist.

Required Parameter

LIB The library that the created objects that contain the system information will be stored in. There is not default value and the library specified must exist on the system.

Example for RTVSYISINF

```
RTVSYISINF LIB(USERLIB)
```

This command will save all of the above information from the system.

No error messages. >>

RTVSMGOBJ (Retrieve System Manager Object) Command Description

Note: To use this command, you must have the 5722-SM1 (System Manager for iSeries) licensed program installed.

RTVSMGOBJ Command syntax diagram

Purpose

The Retrieve System Manager Object (RTVSMGOBJ) command provides the capability of retrieving an OS/400 object from a managed system.

Note: A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

Restrictions:

1. The object cannot reside in the QTEMP library.
2. An object can only be retrieved from a single managed system.

Required Parameters

OBJ Specifies the qualified name to be retrieved.

The possible library values are one of the following:

***CURLIB:** The current library for the job is used to locate the object.

library-name: Specify only the library named in this parameter is searched.

object-name: Specify only characters A through Z, and 0 through 9 are used in the object name.

OBJTYPE

Specifies the object to be used.

object-type: Specify the object type. If *FILEDATA object type is specified, a file member is retrieved without the file attributes.

Optional Parameters

MBR Specifies the OS/400 physical file member name. It is ignored unless the object type is *FILE or *FILEDATA.

***ALL**: The entire physical file is to be retrieved. *ALL must be used if the file is not a physical file.

***FIRST**: Only the first member is retrieved. The member name is determined when the activity is run.

***LAST**: Only the last member is retrieved. The member name is determined when the activity is run by the date the member was added to the physical file.

member-name: Specify the name of the member to be retrieved.

CPNAME

Specifies the APPN control point name of the managed system from which this request is to be retrieved.

***NETATR**: The network ID of the local system is used. This is useful when the node specified is in the same network as the local system.

network-identifier: Specify the managed system APPN network identifier on which the request is to be performed.

control-point-name: Specify the managed system APPN control point name on which the request is to be performed.

TGTRLS

Specifies the release of the operating system on which you intend to use the object.

***CURRENT**: The object is used on the release of the operating system currently running on your system. If V5R2M0 is running on your system, *CURRENT means that you intend to use the object on a system with V5R2M0 installed. The object can also be used on a system with any later release of the operating system installed.

***PRV**: The object is intended for a system which is at the previous release level compared to the local system.

release-level: Specify the release level in the VxRxMx format. The object is used on a system with the specified release or with any later release of the operating system installed.

Valid values depend on the current version, release, and modification level and they change with each new release.

DTACPR

Specifies that data is compressed when retrieving. Compression is valid only for object type *FILEDATA. SNA compression with a blank prime compression character is performed.

***SNA**: The file data is compressed when retrieved.

***NONE**: The file data is not compressed when retrieved.

REPLACE

Specifies if the object should be replaced if it already exists.

***NO:** An error is returned if the object already exists.

***YES:** The object is replaced if it already exists.

Example for RTVSMGOBJ

```
RTVSMGOBJ OBJ(DEVLLIB/NEWPGM) OBJTYPE(*PGM)
          CPNAME(*NETATR DEVSYS)
```

This example shows how to retrieve a program from the development system.

Error messages for RTVSMGOBJ

*ESCAPE Messages

None <<

RTVSYVAL (Retrieve System Value) Command Description

RTVSYVAL Command syntax diagram

Purpose

The Retrieve System Value (RTVSYVAL) command is used in a CL program or REXX procedure to retrieve the value from the specified system value. The value is returned (copied) to the specified CL variable in the program.

Restrictions:

1. The attributes of the system value and the receiving CL variable must be compatible.
2. You must have *ALLOBJ and *SECADM special authorities to change security related system values.

Required Parameters

SYSVAL

Specifies the name of the system value whose value is retrieved and returned for use in the program. The names and descriptions of the system values that can be specified are in the Work

Management  book.

RTNVAR

Specifies the name of the CL program variable that receives the system value being returned. The type and length for the CL variable when it was declared must be compatible with that of the system value being received. The attributes of individual system values are described in the Work

Management  book.

In general, the type of the return variable must match the type of the system value. For character system values that are 1 character long, the CL variable can be a character or logical variable. For character and logical system values, the length of the CL variable must equal the length of the system value. For decimal system values, the CL variable must have a length that is greater than or equal to the length of the system value.

Example for RTVSYVAL

```
RTVSYVAL SYSVAL(QDATE) RTNVAR(&DATE)
```

This command retrieves the date value from the system value QDATE and copies it into the CL variable &DATE. The CL variable must be declared as a 6-character variable to match the attributes of the QDATE system value.

Error messages for RTVSYVAL

***ESCAPE Messages**

CPF1028

&1 not valid for parameter SYVAL.

CPF1074

SYVAL(QMONTH) not valid for Julian date format.

CPF1094

CL variable not same type as system value &1.

CPF1095

CL variable length not valid for system value &1.

CPF1842

Cannot access system value &1.

CPF268D

Unable to access system value &1.

RTVS36A (Retrieve System/36 Attributes) Command Description

RTVS36A Command syntax diagram

Purpose

The Retrieve System/36 Attributes (RTVS36A) command is used in a CL program or REXX procedure to retrieve specific attribute information about the System/36 environment configuration and provide that information to a specified variable of a CL program or REXX procedure. Information about one or more attributes can be retrieved.

More information about the System/36 attributes that can be retrieved is in the description for the Change System/36 Attributes (CHGS36A) command.

Required Parameter

ENV Specifies the name of the System/36 environment from which you are retrieving attributes. The value is #LIBRARY and cannot be changed.

Optional Parameters

SLIB Specifies the name of an 8-character variable to receive the name of the default session library for jobs running in the System/36 environment.

FLIB Specifies the name of a 10-character variable to receive the name of the default files library for jobs running in the System/36 environment.

LIBL Specifies the name of a 4-character variable to receive information on whether the library list is used for jobs running in the System/36 environment. A value of *YES or *NO is returned in the variable.

DATDIFF

Specifies the name of a 4-character variable to receive information on whether files with the same name but different dates can be used for jobs running in the System/36 environment. A value of *YES or *NO is returned in the variable.

S36ESHARE

Specifies the name of a 4-character variable to receive information on whether programs share an open data path (ODP) to database files opened in the System/36 environment. A value of *YES or *NO is returned in the variable.

RCDBLK

Specifies the name of a 4-character variable to receive information on whether record blocking is used for sequential database files sharing an open data path in the System/36 environment. A value of *YES or *NO is returned in the variable.

CACHEDLTF

Specifies the name of a 4-character variable to receive information on whether deleted files are stored in a cache in the System/36 environment. A value of *YES or *NO is returned in the variable.

LPPAGE

Specifies the name of a 3-character variable to receive the number of lines printed on a page for jobs running in the System/36 environment. A value ranging from 1 through 112 is returned in the variable.

FORMTYPE

Specifies the name of a 4-character variable to receive the form type of the printer form used when printing a job in the System/36 environment. A value of *STD or a user-defined form type is returned in the variable.

DFTMSGACN

Specifies the name of a 9-character variable to receive the default message action used by the System/36 environment when an error occurs during the running of a CL command within a System/36 environment procedure. A value of *CONTINUE, *HALT, IGNORE, or *CANCEL is returned in the variable.

HALTOPT

Specifies the name of a 4-character variable to receive the options list for continuation after an error occurs in the System/36 environment and *HALT is specified for the default message action.

EVKJOBINIT

Specifies the name of a 6-character variable to receive the value for the method used to start System/36 EVOKE jobs or job steps in the System/36 environment. A value of *IMMED or *JOBQ is returned in the variable.

EVKJOBPOL

Specifies the name of an 8-character variable to receive the value for the storage pool used for jobs started with the *IMMED option in the System/36 environment. A value of *BASE or *CURRENT is returned in the variable.

EVKJOBPTY

Specifies the name of a 10-character variable to receive the value for the priority at which a job is started when it is started with the *IMMED option in the System/36 environment. A value ranging from 1 through 99 or the value *SUBMITTER is returned in the variable.

SRCRCLEN

Specifies the name of a 3-character variable to receive the record length in bytes for System/36 source files QS36PRC and QS36SRC. A value ranging from 52 through 132 is returned in the variable.

CHGACT

Specifies the name of a 4-character CL variable to receive the value indicating whether the configuration object can be updated using the Change System/36 (CHGS36) command while others are signed on to the System/36 environment. A value of *NO or *YES is returned in the variable.

ADDS36ONLY

Specifies the name of a 4-character CL variable to receive the value indicating whether workstation devices should be added to the System/36 environment configuration when the device signs on to the System/36 environment. A value of *NO or *YES is returned in the variable.

ICFSUBST

Specifies the name of a 4-character CL variable to receive the value on whether to scan intersystem communications function (ICF) start requests for substitution expressions. A value of *NO or *YES is returned in the variable.

MRTUSRPRF

Specifies the name of an 8-character variable to receive the user profile under which the Multiple Requester Terminal (MRT) program is running. A value of *OWNER or *FRSTUSR is returned in the variable.

MRTAUT

Specifies the name of an 8-character variable to receive the user authority to files used by the MRT program. A value of *ALLUSR or *FRSTUSR is returned in the variable.

MRTDLY

Specifies the name of a 5-character variable to receive the time in seconds that the system delays before ending the MRT program. A value ranging from 0 through 32767 is returned in the variable.

MRTJOBINIT

Specifies the name of a 6-character variable to receive the value for the method used to start an MRT job in the System/36 environment. A value of *IMMED or *JOBQ is returned in the variable.

MRTJOBPOL

Specifies the name of an 8-character variable to receive the value for the storage pool to be used for an MRT job started with the *IMMED option in the System/36 environment. A value of *BASE or *CURRENT is returned in the variable.

MRTJOBPTY

Specifies the name of a 10-character variable to receive the value for the priority to start an MRT job started with the *IMMED option. A value ranging from 1 through 99 or the value *SUBMITTER is returned in the variable.

Example for RTVS36A

```
RTVS36A RCBBLK(&RBLOCK) HALT(&OPTION) MRTUSRPRF(&USERID)
```

This command retrieves the shared file record blocking value, the halt options list, and the user profile under which the MRT is running. The file record blocking value is copied into the CL variable &RBLOCK, which must be 4 characters in length. The halt options list is copied into the CL variable &OPTION, which must be at least 4 characters in length. The user profile under which the MRT is running is copied into the CL variable &USERID, which must be 8 characters in length.

No error messages.

RTVTBLSRC (Retrieve Table Source) Command Description

RTVTBLSRC Command syntax diagram

Purpose

The Retrieve Table Source (RTVTBLSRC) command is used to retrieve the source for a *CVT or *SRTSEQ table. These source statements are placed into a source file member, which can be used as input when creating a table with the CRTTBL command. Additionally, this command will convert the source of a *SRTSEQ table to the format used to create a *UCSSRTSEQ table.

Required Parameters

TBL Specifies the qualified name of the table whose source is being retrieved.

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

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

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

library-name: Specify the name of the library to be searched.

table-name: Specify the name of the table whose source is being retrieved.

SRCFILE

Specifies the qualified name of the previously created database source file into which the table source is being written.

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

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

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

library-name: Specify the name of the library to be searched.

source-file-name: Specify the name of the database source file to contain the retrieved table source.

Optional Parameters

SRCMBR

Specifies the name of the database source file member into which the table source is being written. If not specified, the table name is assumed. If the member existed before running the command, it is cleared before any source statements are written into it. If the member did not exist, it is created.

***TBL:** The name of the table is used as the member name.

source-member-name: Specify the name of the source file member to contain the retrieved table source.

CVTTOUCS

Specifies whether the source of a *SRTSEQ table should be converted when placed in the file from the format of a *SRTSEQ table to that of a *UCSSRTSEQ. You would want to do this if you want to create a *UCSSRTSEQ but want to base it on an existing *SRTSEQ table.

***NO:** means that the information is not converted when placed in the file

***YES:** means that the information is converted when placed in the file

Example for RTVTBLSRC

```
RTVTBLSRC  TBL(QSYS/TABLE1) SRCFILE(TBLSRC)
           SRCMBR(TABLEOUT)  CVTTOUCS(*NO)
```

This command retrieves the source statements from the table named TABLE1 in library QSYS. The retrieved source is placed into the file named TBLSRC and is named as member TABLEOUT and not converted to a *UCSSRTSEQ format.

No error messages.

RTVUSRPRF (Retrieve User Profile) Command Description

RTVUSRPRF Command syntax diagram

Purpose

The Retrieve User Profile (RTVUSRPRF) command is used in a CL program or REXX procedure to retrieve one or more of the values stored and associated with a user. The values are returned in the specified variables for the desired user.

The CL prompt for this command lists the minimum length for the variables next to the appropriate parameters to be retrieved. A single digit is shown for character variables. Two digits are shown for decimal variables, where the first digit indicates the minimum variable length, and the second digit indicates the minimum number of decimal positions.

Restriction: *READ authority is required for the user specified on the USRPRF parameter.

Optional Parameters

USRPRF

Specifies the name of the user whose information is being retrieved. If a variable is specified, it must be 10 characters long and contain a user name or the value *CURRENT.

***CURRENT:** The user profile that is currently running is used.

user-name: Specify the name of the user whose information is being retrieved.

RTNUSRPRF

Specifies the name of a variable used to contain the name of the user profile whose information is retrieved. In CL programs, the variable has a length of 10 characters. If *CURRENT is specified on the USRPRF parameter, the value returned is the user profile name of the currently running profile. If a name is specified, that name is returned for this parameter.

SPCAUT

Specifies the name of a variable used to retrieve the list of user special authorities. In CL programs, the variable has a length of 100 characters. The format returned is a list of up to 10 special authority entries, with each entry 10 characters long. If there are fewer than 10 special authorities in the list, the remaining entries are padded on the right with blanks. If the user has no special authorities, the first entry contains the value of *NONE followed by blanks.

The special authority list is returned in the following format:

Entry-1

Special authority CHAR(10)

Entry-2

Special authority CHAR(10)

·
·
·

Entry-10

Special authority CHAR(10)

MAXSTG

Specifies the name of a variable used to retrieve the maximum amount of auxiliary storage space that can be assigned to store permanent objects owned by the specified user. The value returned is either an 11-digit value or a value of -1 if it is *NOMAX. The variable must be an 11-digit value with zero decimal positions (11 0). For additional information on this parameter, see the Create User Profile (CRTUSRPRF) command.

STGUSED

Specifies the name of a variable that is used to retrieve the amount of auxiliary storage currently being used to store permanent objects owned by the specified user profile. In CL programs, the variable has a length of (15 0) characters. The value is returned in kilobytes (1 kilobyte (KB) equals 1024 bytes).

PTYLMT

Specifies the name of a variable used to retrieve the highest scheduling priority the user is allowed to have for each job submitted to the system. In CL programs, the variable has a length of 1 character. This value controls the job processing priority that any job running under this user can have. This means that values specified in the JOBPTY and OUTPTY parameters of any job command cannot exceed the PTYLMT value specified for the user under which the job is run. The scheduling priority can have a value ranging from 0 through 9, where 0 is the highest priority and 9 is the lowest priority.

INLPGM

Specifies the name of a variable used to retrieve the name of the initial program that starts when the specified user signs on the system. In CL programs, the variable has a length of 10 characters. If no initial program name is associated with the specified user, the value returned in the CL variable is *NONE. For more information on this parameter, see the Create User Profile (CRTUSRPRF) command.

INLPGMLIB

Specifies the name of a variable used to retrieve the name of the library that contains the initial program associated with the specified user. In CL programs, the variable has a length of 10 characters. If no initial program is associated with the specified user, blanks are returned in the variable.

JOB Specifies the name of a variable used to retrieve the name of the job description associated with the specified user. In CL programs, the variable has a length of 10 characters.

JOBDLIB

Specifies the name of a variable used to retrieve the name of the library that contains the job description associated with the specified user. In CL programs, the variable has a length of 10 characters.

GRPPRF

Specifies the name of a variable used to retrieve the name of the group profile. In CL programs, the variable has a length of 10 characters. If no group profile exists for the specified user profile, a value of *NONE is returned in the variable. For more information on this parameter, see the CRTUSRPRF (Create User Profile) command.

OWNER

Specifies the name of a variable used to retrieve the special value of *USRPRF or *GRPPRF. In CL programs, the variable has a length of 10 characters. This parameter indicates the owner of newly-created objects. This is either the specified user or the user's group profile. If no group profile exists for the specified user profile, the value returned in the variable is *USRPRF.

GRPAUT

Specifies the name of a variable used to retrieve the authority granted to the group profile for

newly-created objects. In CL programs, the variable has a length of 10 characters. The special value of *NONE, *CHANGE, *ALL, *USE, or *EXCLUDE is returned in the variable. If there is no group profile for the specified user, the special value of *NONE is returned. If the group profile is the owner of the objects created by the specified user, the special value returned is *NONE. For more information on this parameter, see the CRTUSRPRF (Create User Profile) command.

ACGCDE

Specifies the name of a variable used to retrieve the value of the job accounting code assigned to the specified user. In CL programs, the variable has a length of 15 characters. If no job accounting code exists for the user profile, blanks are returned to the user. More information on this parameter is in the Create User Profile (CRTUSRPRF) command description.

MSGQ

Specifies the name of the CL variable that receives the group message queue name. This must be a character variable with a minimum length of 10 characters. If the message queue name has fewer characters than the variable allows, the value is padded on the right with blanks.

MSGQLIB

Specifies the name of a variable used to retrieve the name of the library containing the message queue associated with the specified user. In CL programs, the variable has a length of 10 characters.

OUTQ Specifies the name of a variable used to retrieve the name of the output queue associated with the specified user. In CL programs, the variable has a length of 10 characters. The special value *DEV or *WRKSTN is returned in the variable.

OUTQLIB

Specifies the name of a variable used to retrieve the name of the library containing the output queue associated with the specified user. In CL programs, the variable has a length of 10 characters. Blanks are returned if the current value for OUTQ is *DEV or *WRKSTN.

TEXT Specifies the name of a variable used to retrieve the text description of the user profile. In CL programs, this should be a 50-character variable. If there is no text associated with the user profile, blanks are returned in the CL variable.

PWDCHGDAT

Specifies the name of a variable used to retrieve the date of the last time the password for the specified user was changed. In CL programs, the variable has a length of 6 characters. The date is returned in the form YYMMDD. If the user does not have a password change date, blanks are returned.

USRCLS

Specifies the name of a variable used to retrieve the user class for the specified user. In CL programs, the variable has a length of 10 characters. The special value of *USER, *SYSOPR, *PGMR, *SECADM, or *SECOFR is returned in the variable.

ASTLVL

Specifies the assistance-level. The special value of *SYSVAL, *BASIC, *INTERMED, or *ADVANCED is returned as the variable.

SPCENV

Specifies the name of a variable for the starting environment for the specified user. In CL programs, the variable has a length of 10 characters. The special value of *SYSVAL, *NONE, or *S36 is returned in the variable.

CURLIB

Specifies the name of the variable used to retrieve the name of the job's default library for the specified user. In CL programs, the variable has a length of 10 characters. A value of *CRTDFT is returned in the variable if no current library exists for this user.

INLMNU

Specifies the name of a variable used to retrieve the name of the user's first menu when the

specified user signs on the system. In CL programs, the variable has a length of 10 characters. For more information on this parameter, see the CRTUSRPRF (Create User Profile) command.

INLMNULIB

Specifies the name of the variable used to retrieve the library name containing the initial menu. In CL programs, the variable has a length of 10 characters.

LMTCPB

Specifies the name of a variable used to retrieve the value for the limits to which users can change their user profiles and run commands. In CL programs, the variable has a length of 10 characters. The special value of *NO, *YES, or *PARTIAL is returned in the variable.

DLVRY

Specifies the name of a variable used to retrieve the message control delivery value for the specified user profile. In CL programs, the variable has a length of 10 characters. The special value of *NOTIFY, *BREAK, *HOLD, or *DFT is returned in the variable.

SEV Specifies the name of a variable used to retrieve the message control severity level for the specified user. In CL programs, the variable has a length of (2 0) characters.

PRTDEV

Specifies the name of a variable used to retrieve the name of the printer device for the specified user. In CL programs, the variable has a length of 10 characters. A value of *SYSVAL is returned if the printer device name is from the system value QPRTDEV. A value of *WRKSTN is returned if the printer device name is from the printer device assigned to the user's work station.

ATNPGM

Specifies the name of a variable used to retrieve the name of the ATTN key handling program for the specified user. In CL programs, the variable has a length of 10 characters. A value of *SYSVAL or *NONE is returned.

ATNPGMLIB

Specifies the name of a variable used to retrieve the name of the library containing the ATTN key handling program for the specified user. In CL programs, the variable has a length of 10 characters. If *NONE is the current value of ATNPGM, blanks are returned in the variable.

USROPT

Specifies the name of a variable used to retrieve the list of user option values for the specified user. In CL programs, the variable has a length of 240 characters. The special value *NONE or a list of values is returned in the variable.

DSPSGNINF

Specifies the name of a variable used to retrieve the display sign-on information display indicator for the specified user. In CL programs, the variable has a length of 7 characters. The special value of *SYSVAL, *YES, or *NO is returned in the variable.

PWDEXPITV

Specifies the name of a variable used to retrieve the password expiration interval for the specified user. In CL programs, the variable specified must be packed (5 0) in length. The value returned is either a value ranging from 1 through 366, 0, if it is *SYSVAL, or -1 if it is *NOMAX.

PWDEXP

Specifies the name of a variable used to retrieve the password expired indicator for the specified user. In CL programs, the variable has a length of 4 characters. The special value of *YES or *NO is returned in the variable.

STATUS

Specifies the name of a variable used to retrieve the status of the specified user. In CL programs, the variable has a minimum length of 10 characters. The special value of *ENABLED or *DISABLED is returned.

PRVSIGN

Specifies the name of a variable used to retrieve the previous sign-on date and time for the specified user. In CL programs, the variable has a length of 13 characters. The date and time are returned in the form CYYMMDDHHMMSS. If the user does not have a previous sign-on date and time, blanks are returned.

NOTVLDSIGN

Specifies the name of a variable used to retrieve the number of sign-on attempts that were not valid for the specified user. In CL programs, the variable specified must be packed (11 0) in length.

LMTDEVSSN

Specifies the name of a variable used to retrieve the limit device sessions indicator for the specified user. In CL programs, the variable has a length of 7 characters. The special value of *SYSVAL, *YES, or *NO is returned in the variable.

KBDBUF

Specifies the name of a variable used to retrieve the keyboard buffering value for the specified user. In CL programs, the variable has a length of 10 characters. The special value of *SYSVAL, *NO, *TYPEAHEAD, or *YES is returned in the variable.

LANGID

Specifies the name of a variable that is used to retrieve the language identifier for the specified user. The value returned is either *SYSVAL or the 3-character language identifier. If *SYSVAL is returned, the language identifier for the user is determined by the QLANGID system value. In CL programs, the variable has a length of 10 characters.

CNTRYID

Specifies the name of a variable used to retrieve the country or region identifier for the specified user. The value returned is either *SYSVAL or the 2-character country or region identifier. If *SYSVAL is returned, the country or region identifier for the user is determined by the QCNTRYID system value. In CL programs, the variable has a length of 10 characters.

CCSID

Specifies the name of a variable used to retrieve the coded character set identifier (CCSID) for the specified user. The value returned is either a 5-digit value or a value of -2 if it is *SYSVAL. If *SYSVAL is returned, the CCSID for the user is determined by the QCCSID system value. In CL programs, this should be a decimal variable length of (5 0).

SRTSEQ

Specifies the name of a variable used to retrieve the sort sequence table for the specified user. The value returned is one of the following: *HEX, *LANGIDUNQ, *LANGIDSHR, *SYSVAL, or the 10-character table identifier. If *SYSVAL is returned, the table identifier for the user is determined by the QSRTSEQ system value.

SRTSEQLIB

Specifies the name of a variable used to retrieve the sort sequence table library for the specified user. The value returned is the 10-character library identifier. The variable is set to blanks unless a sort sequence table name is specified.

OBJAUD

Specifies the name of a 10-character variable used to retrieve the object auditing value for the specified user. The special value of *NONE, *CHANGE, or *ALL, as specified on the Change User Audit (CHGUSRAUD) command, is returned in the variable. When less than 10 characters are returned, the variable is padded on the right with blanks.

AUDLVL

Specifies the name of a variable used to retrieve the object auditing level for the specified user. In CL programs, the variable has a length of 640 characters. The format returned is a list of a maximum of 64 object auditing level entries, with each entry 10 characters long. If there are fewer than 64 object auditing level entries in the list, the remaining entries are padded on the right with blanks. If the user has no object auditing levels, the first entry contains the value of *NONE

followed by blanks. If the user has object auditing levels, one or more of the following special values, as specified in the Change User Audit (CHGUSRAUD) command, is returned in the variable: *CMD, *CREATE, *DELETE, *JOBDDTA, *OBJMGT, *OFCSRV, *PGMADP, *SAVRST, *SECURITY, *SERVICE, *SPLFDDTA, or *SYSMGT.

The object auditing level list is returned in the following format:

Entry-1

Object auditing level CHAR(10)

Entry-2

Object auditing level CHAR(10)

.
.
.

Entry-64

Object auditing level CHAR(10)

GRPAUTYP

Specifies the name of a variable used to retrieve the type of authority to be granted to the group profile for newly-created objects. In CL programs, the variable has a length of 10 characters. The special value of *PRIVATE or *PGP is returned in the variable.

SUPGRPPRF

Specifies the name of a variable used to retrieve the supplemental group profiles for the specified user. In CL programs, the variable has a length of 150 characters. The format returned is a list of up to 15 supplemental group profiles, with each entry 10 characters long. If there are fewer than 15 supplemental groups in the list, the remaining entries are padded on the right with blanks. If the user has no supplemental groups, the first entry contains the value of *NONE followed by blanks.

The supplemental group profiles are returned in the following format:

Entry-1

Supplemental group profile CHAR(10)

Entry-2

Supplemental group profile CHAR(10)

.
.
.

Entry-15

Supplemental group profile CHAR(10)

UID Specifies the name of a variable used to retrieve the user ID number (uid) for the specified user. In CL programs, this should be a decimal variable length of 10 0.

GID Specifies the name of a variable used to retrieve the group ID number (gid) for the specified user. In CL programs, this should be a decimal variable length of 10 0. The value returned is either a 10-digit value or a value of 0 if the gid number is *NONE. The variable must be a 10-digit value with no decimal positions.

SETJOBATR

Specifies the name of a variable used to retrieve the job attributes set from the LOCALE path name of the specified user. In CL programs, the variable has a length of 160 characters. The format returned is a list of up to 16 job attributes that are set, with each entry 10 characters long. If there are fewer than 16 attributes in the list, the remaining entries are padded on the right with blanks. If no attributes are set from the locale path name for the user, the first entry contains the value of *NONE followed by blanks.

CHRIDCTL

Specifies the name of a variable used to retrieve the character identifier control for the specified user. In CL programs, this variable should have a length of 10 characters. The special values of *DEV, *JOBCCSID, or *SYSVAL are returned in the variable. If *SYSVAL is returned, the CHRID control for the user is determined by the QCHRIDCTL system value.

Example for RTVUSRPRF

```
CRTUSRPRF  USRPRF(SMITH)  SPCAUT(*SAVSYS *SECADM)
           MAXSTG(*NOMAX) PTYLM(4)  INLPGM(*NONE)
           MSGQ(QGPL/SMITHMQ)  OUTQ(QGPL/QSMITH)
           TEXT('John Smith User Profile')
```

The user specifies the above command.

When user Smith calls a CL program containing the following:

```
DCL  &UNAME      *CHAR 10
DCL  &URIGHT     *CHAR 100
DCL  &IPGM       *CHAR 10
DCL  &IPGMLB    *CHAR 10
DCL  &UMSGQ     *CHAR 10
DCL  &UMSQLB    *CHAR 10
DCL  &USED      *CHAR 10

RTVUSRPRF  USRPRF(*CURRENT)  STGUSED(&USED)
           RTNUSRPRF(&UNAME)  SPCAUT(&URIGHT)  INLPGM(&IPGM)
           INLPGMLIB(&IPGMLB)
```

The following values are returned:

```
&UNAME      'SMITH'
&URIGHT     '*SAVSYS *SECADM (          ) ...'
            |_____100 characters_____|
&IPGM       '*NONE'
&IPGMLB     ''
```

Error messages for RTVUSRPRF

*ESCAPE Messages

CPF2203

User profile &1 not correct.

CPF2204

User profile &1 not found.

CPF2213

Not able to allocate user profile &1.

CPF2217

Not authorized to user profile &1.

CPF2225

Not able to allocate internal system object.

CPF8134

User profile &4 damaged.

RTVWSCST (Retrieve Work Station Customizing Object Source)

Command Description

RTVWSCST Command syntax diagram

Purpose

The Retrieve Work Station Customizing Object (RTVWSCST) command allows the user to retrieve a system-supplied set of table attributes for a given device type, keyboard language type, and keyboard type or a given manufacturer, type, and model of an ASCII printer into a source physical file member.

Required Parameters

DEVTYPE

Specifies the device type.

***TRANSFORM:** The SCS-to-ASCII host print transform function support is used by the ASCII printer.

device-type: Specify the device type to be used. See the Workstation Customization Programming



book for a list of allowed device types.

MFRTYPMDL

Specifies the manufacturer, type, and model for an ASCII printer using host print transform function support. See the Create Device Description (Printer) (CRTDEVPRT) command for a list of the supported manufacturers, types, and models for ASCII printers using host print transform function support.

KBDTYPE

Specifies the 3-character country or region keyboard language identifier (used for EBCDIC and ASCII) for this display station.

See the KBDTYPE parameter on the Change Device Description (Display) (CHGDEVDS) or the Create Device Description (Display) (CRTDEVDS) for a list of the valid identifiers and the languages the identifiers represent. The ASCII device groups (if applicable) are also shown for each language.

SRCMBR

Specifies the name of the source file member to receive the retrieved table attributes.

Optional Parameters

KBD Specifies the keyboard type.

***DATA5250:** A 5250 data entry keyboard is specified.

***TYPE5250:** A 5250 typewriter keyboard is specified.

***DATA122:** A 122 key data entry keyboard is specified.

***TYPE122:** A 122 key typewriter keyboard is specified.

***ENHANCED:** An enhanced keyboard is specified.

SRCFILE

Specifies the name of the source file in which a member is created to contain the retrieved table attributes. If the source file does not exist, it is created. The coded character set identifier for the source file is *HEX.

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

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

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

library-name: Specify the name of the library to be searched.

QXTSRC: The IBM-supplied source file QXTSRC is used.

source-file-name: Specify the name of the source file.

TEXT Specifies the text that briefly describes the source physical file member. More information on this parameter is in Commonly used parameters.

***BLANK:** Text is not specified.

'description': Specify a description for the source physical file member.

Example for RTVWSCST

```
RTVWSCST  DEVTYPE(5251)  KBDTYPE(USB)  SRCMBR(MYSOURCE)
          KBD(*DATA5250)  SRCFILE(MYLIB/QXTSRC)
```

This command retrieves the system mapping tables for a 5251 twinaxial display with a 5250 data entry type keyboard attached using the U.S. basic language. The tables are stored in source member MYSOURCE in source file QXTSRC in library MYLIB.

Error messages for RTVWSCST

*ESCAPE Messages

CPF5D33

Request for system table failed. Reason code &4.

RETURN (Return) Command Description

RETURN Command syntax diagram

Purpose

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 (PWRDWNSYS) 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.

Example for RETURN

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.

No error messages.

RVKACCAUT (Revoke Access Code Authority) Command Description

RVKACCAUT Command syntax diagram

Purpose

The Revoke Access Code Authority (RVKACCAUT) command takes away the access code authority for an individual user or a group of users.

Note:

You can specify USER(*CURRENT) to remove access code authority from any access code to which you have authority.

Restriction: You must have *ALLOBJ authority to take away access code authority for other users.

Required Parameters

ACC Specifies the access code for which authority is revoked. The access code is a decimal number ranging from 1 through 2047.

***ALL:** All access code authority for the user is revoked.

access-code: Specify a decimal number, ranging from 1 through 2047, that specifies the access code authority that is revoked.

USER Specifies the name of the user profile for whom access code authority is revoked.

***CURRENT:** The user profile that is currently running is used.

***ALL:** Access code authority is revoked for all users.

user-profile-name: Specify the profile name of the user whose access code authority is revoked.

Examples for RVKACCAUT

Example 1: Revoking Authority of Current User

```
RVKACCAUT ACC(250) USER(*CURRENT)
```

This command takes away the access code authority of access code 250 from the user currently running this command.

Example 2: Revoking Authority of Specific User

```
RVKACCAUT ACC(300) USER(BILLY)
```

This command takes away the access code authority of access code 300 from user BILLY. This command was run by someone with *ALLOBJ or *SECADM special authority, or by BILLY. A user who runs this command for himself can enter USER(*CURRENT) or his own user profile name; they are the same.

Error messages for RVKACCAUT

*ESCAPE Messages

CPF9009

System requires file &1 in &2 be journaled.

CPF9014

Access code authority removed from &1 users, not revoked from &2 users.

CPF9024

System cannot get correct record to finish operation.

CPF9037

Not allowed to specify USER(*ALL).

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

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

RVKOBJAUT (Revoke Object Authority) Command Description

RVKOBJAUT Command syntax diagram

Purpose

The Revoke Object Authority (RVKOBJAUT) command is used to take away specific (or all) authority for the named objects from one or more users also named in the command, or to remove the authority of an authorization list for the named objects. This command can be entered by the security officer, by an object's owner, or by a user who has object management authority for the object being removed. A user with object management authority can remove only the authorities that that user has. A user may not be able to give or remove authorities for an object that has been allocated (locked) to another job. If a specific (not *ALL) authority cannot be revoked, a message is issued that indicates the authorities that were not revoked.

Restrictions:

1. Before this command is used to remove authorities to use a device, control unit, or line description, its associated device, control unit, or line must be varied on.

2. Authority to use a device cannot be revoked if a user is currently signed on to the device.

Note:

Users can revoke their own authority to a device if they are currently signed onto that device. However, doing so may produce unpredictable results and is not advisable.

3. For display stations or for work station message queues associated with the display station, if this command is not entered at the device for which authorities are being revoked, it should be preceded by the Allocate Object (ALCOBJ) command and followed by the Deallocate Object (DLCOBJ) command.
4. Object type *DOC or *FLR cannot be specified. Document interchange support must be used.
5. Object type *AUTL cannot be specified. The Change Authorization List Entry (CHGAUTLE) or Remove Authorization List Entry (RMVAUTLE) commands must be used.
6. AUT (*AUTL) can be specified only with USER (*PUBLIC).
7. Only a user with *ALL authority or the owner can remove the authorization list.
8. >> You must have *USE authority to the auxiliary storage pool device if one is specified.<<

Security Risk

Revoking all authorities specifically given to a user for an object can result in the user having more authority than before the revoke operation. If a user has *USE authority for an object and *CHANGE authority on the authorization list that secures the object, revoking *USE authority results in the user having *CHANGE authority to the object.
--

Required Parameters

OBJ Specifies the qualified name of the objects for which specific authority is revoked. Either a specific or a generic object name is specified with a library name. If *ALL is specified, the name of a library must be specified. For more information on the use of generic functions, refer to generic functions.

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

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

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

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

***ALL:** >> All libraries in the auxiliary storage pools (ASPs) specified by the ASPDEV parameter are searched. <<

***ALLUSR:** >> All user libraries in the ASPs specified by the ASPDEV parameter are searched. << All libraries with names that do not begin with the letter Q are searched except for the following:

#CGULIB
#COBLIB
#DFULIB
#DSULIB

#RPGLIB
#SDALIB
#SEULIB

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

QDSNX	QS36F	QUSROND
QGPL	QUSER38	QUSRPOSGS
QGPL38	QUSRADSM	QUSRPOSSA
QMPGDATA	QUSRBRM	QUSRPYMSVR
QMQMDATA	QUSRDIRCL	QUSRRDARS
QMQMPROC	QUSRDIRDB	QUSRSYS
QPFRDATA	QUSRIJS	QUSRVI
QRCL	QUSRINFSKR	QUSRVxRxMx
» QRCLnnnnn «	QUSRNOTES	

Notes:

1. » “nnnnn” is the number of a primary auxiliary storage pool. «
2. A different library name, of the form QUSRVxRxMx, can be created by the user for each release that IBM supports. VxRxMx is the version, release, and modification level of the library.



***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. «

library-name: Specify the name of the library to be searched.

***ALL**: All objects of the specified type (OBJTYPE) found in the search have specific authorities revoked. Specify the name of a library with *ALL.

generic-object-name*: Specify the generic name of the object. 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 revoked only if *ALL, *ALLUSR, » *ALLAVL, or *ALLUSRAVL « library values can be specified for the name. For more information on the use of generic names, refer to generic names.

object-name: Specify the name of the object for which specific authorities are revoked.

OBJTYPE

Specifies the object type of the object that has specific authorities revoked. More information on this parameter is in Commonly used parameters.

***ALL**: All object types have specific authorities revoked.

object-type: Specify the type of the object for which specific authorities are revoked.

USER Specifies the names of one or more users whose specific authorities to the named object are being removed. If a user was given the authority by USER(*PUBLIC) being specified in the Grant Object Authority (GRTOBJAUT) command, the same authorities are revoked by *PUBLIC being

specified in this parameter. Users who were given specific authority by having their names specified in the GRTOBJAUT command must have their names specified on this parameter to remove the same authorities.

Either this parameter or the AUTL parameter must be specified.

***ALL:** The authorities specified in the AUT parameter are taken away from all enrolled users of the system except the owner, if they are publicly or explicitly authorized.

***PUBLIC:** The specified authorities are taken away from users who do not have specific authority for the object, who are not on the authorization list, and whose group has no authority. Users who have specific authority still retain their authorities to the object.

user-profile-name: Specify the user profile names of one or more users that are having the specified authorities revoked. The authorities specified in the AUT parameter are being specifically taken away from each specified user. This parameter cannot be used to remove public authority from specific users; only authorities that were specifically given to them can be specifically revoked.

AUTL Specifies that the authorization list is revoked from the object specified on the OBJ parameter. If public authority to the object is *AUTL, it is changed to *EXCLUDE.

Either this parameter or the USER parameter must be specified. If this parameter is specified, the AUT parameter is ignored.

Optional Parameters

AUT Specifies the authority being revoked from the users who do not have specific authority to the object, who are not on an authorization list, and whose user group has no specific authority to the object.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the object. If the object is an authorization list, the user cannot add, change, or remove user profile names. Revoking *ALL authority from *PUBLIC causes the *PUBLIC authority to change to *EXCLUDE.

***USE:** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. *USE authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The user cannot access the object.

***AUTL:** The public authority of the authorization list specified in this parameter is revoked for the object. The public authority for the object becomes *EXCLUDE.

A maximum of ten of the following values can be specified:

***OBJALTER:** Object alter authority provides the authority needed to alter the attributes of an object. If the user has this authority on a database file, the user can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user has this authority on an SQL package, the user can change the attributes of the SQL package. This authority is currently only used for database files and SQL packages.

***OBJEXIST:** Object existence authority provides the authority to control the object's existence and ownership. These authorities are necessary for users who want to delete the object, free storage of the object, perform save and restore operations for the object, or transfer ownership of an

object. If a user has special save system authority (*SAVSYS), object existence authority is not needed. Object existence authority is required to create an object that has been named by an authority holder.

***OBJMGT:** Object management authority provides the authority to specify the security for the object, move or rename the object, and add members to database files.

***OBJOPR:** Object operational authority provides authority to look at the description of an object and use the object as determined by the user's data authorities to that object.

***OBJREF:** Object reference authority provides the authority needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user has this authority on a physical file, the user can add referential constraints in which the physical file is the parent. This authority is currently only used for database files.

***ADD:** Add authority provides the authority to add entries to an object (for example, job entries to a queue or records to a file).

***DLT:** Delete authority allows the user to remove entries from an object, for example, remove messages from a message queue or records from a file.

***EXECUTE:** Execute authority provides the authority needed to run a program or to locate an object in a library.

***READ:** Read authority provides the authority needed to show the contents of an entry in the object.

***UPD:** Update authority provides the authority needed to change the entries in the object.

➤ ASPDEV

Specifies the auxiliary storage pool (ASP) device name where the library that contains the object (OBJ parameter) is located. If the object's library resides in an ASP that is not part of the library name space associated with the job, this parameter must be specified to ensure the correct object is used as the target of the revoke operation.

***:** The ASPs that are currently part of the job's library name space will be searched to locate the object. This includes the system ASP (ASP number 1), all defined basic user ASPs (ASP numbers 2-32), and, if the job has an ASP group, all independent ASPs in the ASP group.

***SYSBAS:** The system ASP and all basic user ASPs will be searched to locate the object. No independent ASPs will be searched, even if the job has an ASP group.

auxiliary-storage-pool-device-name: The device name of the independent ASP to be searched to locate the object. The independent ASP must have been activated (by varying on the ASP device) and have a status of 'Available'. The system ASP and basic user ASPs will not be searched. ⏪

Examples for RVKOBJAUT

Example 1: Removing Authority From All Users Except Program Owner

```
RVKOBJAUT OBJ(ARLIB/PROG1) OBJTYPE(*PGM) USER(*ALL)
```

This command removes the authorities (AUT was not specified; *CHANGE is assumed) from all users who were either explicitly or publicly authorized, except the owner, for the program (*PGM) named PROG1 located in the library named ARLIB.

Example 2: Removing Object Owner's Authority to Delete a Program

```
RVKOBJAUT OBJ(TSMITHPGM/MITHLIB)
OBJTYPE(*PGM) USER(TSMITH)
AUT(*OBJEXIST)
```

This command removes the object owner's (TSMITH) authority to delete a program (TSMITHPGM) in his library (SMITHLIB). The object owner might do this to ensure that the object is not deleted by mistake. If the owner ever wants to delete the object, object existence authority for the object can be granted by using the Grant Object Authority (GRTOBJAUT) command).

Example 3: Removing *DLT and *UPD Authorities

```
RVKOBJAUT OBJ(FILEX) OBJTYPE(*FILE)
  USER(HEANDERSON) AUT(*DLT *UPD)
```

This command removes delete and update authorities for the file named FILEX from the user HEANDERSON.

Example 4: Removing *OBJEXIST Authority

```
RVKOBJAUT OBJ(ARLIB/ARJOBDD) OBJTYPE(*JOBDD)
  USER(RLJOHNSON) AUT(*OBJEXIST)
```

This command removes the object existence authority for the object named ARJOBDD from the user RLJOHNSON. ARJOBDD is a job description that is located in the library named ARLIB.

Example 5: Removing Specific Authorities

```
RVKOBJAUT OBJ(FILEX) OBJTYPE(*FILE)
  AUTL(FILEUSERS)
```

This command removes specific authorities for the file named FILEX from the users in the authorization list FILEUSERS.

Error messages for RVKOBJAUT

*ESCAPE Messages

CPF22A0

Authority of *AUTL is allowed only with USER(*PUBLIC).

CPF22A1

OBJTYPE(*AUTL) not valid on this command.

CPF22A2

Authority of *AUTL not allowed for object type *USRPRF.

CPF22A3

AUTL parameter not allowed for object type *USRPRF.

CPF22A4

*EXCLUDE cannot be revoked from *PUBLIC.

CPF22A5

Object &1 in &3 type *&2 not secured by authorization list &4.

CPF22DA

Operation on file &1 in &2 not allowed.

CPF2207

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

CPF2208

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

CPF2209

Library &1 not found.

- CPF2210**
Operation not allowed for object type *&1.
- CPF2211**
Not able to allocate object &1 in &3 type *&2.
- CPF2216**
Not authorized to use library &1.
- CPF2224**
Not authorized to revoke authority for object &1 in &3 type *&2.
- CPF2227**
One or more errors occurred during processing of command.
- CPF2236**
AUT input value not supported.
- CPF2243**
Library name &1 not allowed with OBJ(generic name) or OBJ(*ALL).
- CPF2253**
No objects found for &1 in library &2.
- CPF2254**
No libraries found for &1 request.
- CPF2273**
Authority may not have been changed for object &1 in &3 type *&2 for user &4.
- CPF2283**
Authorization list &1 does not exist.
- CPF326A**
Operation not successful for file &1 in library &2.
- CPF327F**
Operation not successful for file &1 in library &2.
- CPF3381**
Revoke authority not allowed from spool user profile QSPL.
- » **CPF980B**
Object &1 in library &2 not available. «
- CPF9804**
Object &2 in library &3 damaged.
- » **CPF9814**
Device &1 not found.
- CPF9825**
Not authorized to device &1.
- CPF9873**
ASP status is preventing access to object. «

RVKPUBAUT (Revoke Public Authority) Command Description

RVKPUBAUT Command syntax diagram

Purpose

The Revoke Public Authority (RVKPUBAUT) command limits the use of a set of IBM-supplied commands and programs by changing the public authority to *EXCLUDE.

This command can be customized by the security administrator by following the steps below:

1. Issue the Retrieve CL Source (RTVCLSRC) command against the QSECRVKP program.
2. Edit the source code produced from the RTVCLSRC command and compile the new program. Make sure that the program is given a new name, is created into a library other than QSYS, and that the *PUBLIC authority is set to *EXCLUDE.
3. Issue the Change Command (CHGCMD) against the Revoke Public Authority (RVKPUBAUT) command and specify your new program for the **PGM** parameter. An example is listed below.

```
CHGCMD  CMD(QSYS/RVKPUBAUT)
        PGM(library-name/new-pgm-name)
```

Note:

If a product upgrade is done, if the RVKPUBAUT command is reinstalled, or if maintenance is applied to the RVKPUBAUT command, the CHGCMD will have to be issued again to customize the command.

Restrictions:

You must have *ALLOBJ special authority to use this command.

Optional Parameter

LIB Specifies the name of the library where the IBM supplied commands are to be found.

Note:

The library value is only used for commands because they can be located in secondary language libraries.

QSYS: Specifies the commands are located in library QSYS.

library-name: Specify the library where the commands are located.

Example for RVKPUBAUT

```
RVKPUBAUT  LIB(QSYS)
```

The public authority for commands and programs in library QSYS is set to *EXCLUDE.

Error messages for RVKPUBAUT

***ESCAPE Messages**

CPFB304

User does not have required special authorities.

RVKUSRPMN (Revoke User Permission) Command Description

RVKUSRPMN Command syntax diagram

Purpose

The Revoke User Permission (RVKUSRPMN) command takes away from one user (or all users) the permission to access documents and folders on behalf of another user.

Note:

If work is being done on behalf of another user at the time this command is run, any functions that have started are completed; however, additional functions are not accepted.

Restriction: The user must have *ALLOBJ authority to revoke document authority for other users.

Required Parameter**FROMUSER**

Specifies the user profile name of the user whose permission is being revoked.

***ALL:** All users who are currently permitted to work on behalf of another user are no longer permitted.

user-profile-name: Specify the name of the user profile that is no longer permitted to work on behalf of the user specified in the FORUSER parameter.

Optional Parameter**FORUSER**

Specifies the user profile name of the user on whose behalf the user specified in the FROMUSER parameter can no longer work.

***CURRENT:** The user profile that is currently running is used.

user-profile-name: Specify the name of the user profile on whose behalf other users are no longer permitted to work. Up to 50 user profile names can be specified. The user must have *ALLOBJ or *SECADM special authority to specify a user profile other than his or her own.

Example for RVKUSRPMN

```
RVKUSRPMN FROMUSER(JOHNSON) FORUSER(ANDERSON)
```

This command takes away user permission from JOHNSON for ANDERSON. The user JOHNSON is no longer allowed to work on behalf of ANDERSON.

Error messages for RVKUSRPMN***ESCAPE Messages****CPF9008**

User permission revoked for &1 users, not revoked for &2.

CPF9009

System requires file &1 in &2 be journaled.

CPF9024

System cannot get correct record to finish operation.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

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

RVKWSOAUT (Revoke Workstation Object Authority) Command Description

RVKWSOAUT Command syntax diagram

Purpose

The Revoke Workstation Object Authority (RVKWSOAUT) command is used to take away authority for a workstation object used by the OS/400 Graphical Operations program. Specific or all authority can be taken away from one or more users named in the command. You also can take away the authority of an authorization list for a named object.

This command can be issued by a security officer, by an object owner, or by a user who has object management authority to the object for which authority is to be revoked. If a specific authority (other than *ALL) is specified on the AUT parameter, and that authority is not revoked, a message is issued that indicates the authority that is not revoked.

Security Risk
Revoking all authorities specifically given to a user for an object can result in the user having more authority than before the revoke operation. If a user has *USE authority for an object and *CHANGE authority on the authorization list that secures the object, revoking *USE authority results in the user having *CHANGE authority to the object.

Restrictions:

1. If you have object management authority, you can revoke only the explicit authority that you have.
2. You might not be able to revoke authority for an object that has been allocated (locked) to another job. Authority cannot be revoked for an object that is currently in use.

Required Parameters

WSOTYPE

Specifies the name of the workstation object for which specific authorities are to be revoked.

The special values for this parameter are described in the following table.

Special Value	Workstation Objects
*TPLWRKARA	Work area template
*WRKARA	Work area objects
*TPLPRTOL	Printer output list template
*PRTOL	Printer output list objects
*TPLTPRTL	Printer list template
*PRTL	Printer list objects
*TPLOUTQ	Output queue template
*TPLOUTQL	Output queue list template
*OUTQL	Output queue list objects
*TPLJOBL	Job list template
*JOBL	Job list objects
*TPLJOBQ	Job queue template
*TPLJOBLOG	Job log template
*JOBLOG	Job log objects
*TPLJOBQL	Job queue list template
*JOBQL	Job queue list objects
*TPLMSGL	Message list template
*MSGL	Message list objects
*TPLMSGQ	Message queue template
*TPLMSGSND	Message sender template

Special Value

*MSGSEND
*TPLSGNUSL
*SGNUSL
*TPLOBJL
*OBJL
*TPLLBSL
*LIBSL
*TPLLIB
*TPLLAUNCH
*LAUNCH
*PRSSET

Workstation Objects

Message sender
Signed-on user list template
Signed-on user list objects
Object list template
Object list objects
Library list template
Library list objects
Library template
Job submitter template
Job submitter objects
Personal setting objects

USER Specifies the user profile names of one or more users whose specific authorities to the named object are being revoked.

Authorities revoked by this command are related to those given by the Grant Workstation Object (GRTWSOAUT) command. If users have public authority to an object because USER(*PUBLIC) was specified on the GRTWSOAUT command, that public authority is revoked when *PUBLIC is specified on this parameter. If users have specific authorities to an object because their names were specified on the GRTWSOAUT command, their names must be specified on this parameter to revoke the same authorities.

Note:

Either this parameter or the AUTL parameter must be specified.

***ALL:** The authorities specified on the AUT parameter are taken away from all enrolled users of the system except the owner, if they are publicly or explicitly authorized.

***PUBLIC:** The specified authorities are taken away from users who do not have specific authority for the object, who are not on the authorization list, and whose group has no authority. Users who have specific authority still retain their authorities to the object.

user-profile-name: Specify the user profile names of one or more users that are having the specified authorities revoked. The authorities specified in the AUT parameter are being specifically taken away from each specified user. This parameter cannot be used to remove public authority from specific users; only authorities that were specifically given to them can be specifically revoked.

AUTL Specifies the authorization list that is revoked from the object specified on the WSOTYPE parameter. If public authority to the object is *AUTL, it is changed to *EXCLUDE. The authorization list's authority is then removed.

Either this parameter or the USER parameter must be specified. If this parameter is specified, the AUT parameter is ignored.

Optional Parameter

AUT Specifies the authority to be taken away from users who do not have specific authority to the object, who are not on an authorization list, and whose user group has no specific authority to the object.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the workstation object. Revoking *ALL authority from *PUBLIC causes the *PUBLIC authority to change to *EXCLUDE.

***USE:** The user can perform basic operations on the workstation object, such as running a program or reading a file. The user cannot change the workstation object. *USE authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The user cannot access the workstation object.

***AUTL:** The public authority of the authorization list specified in this parameter is revoked for the object. The public authority for the object becomes *EXCLUDE.

A maximum of ten of the following values can be specified:

***OBJALTER:** Object alter authority provides the authority needed to alter the attributes of an object. If the user has this authority on a database file, the user can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user has this authority on an SQL package, the user can change the attributes of the SQL package. This authority is currently only used for database files and SQL packages.

***OBJEXIST:** Object existence authority provides the authority to control the object's existence and ownership. These authorities are necessary for users who want to delete the object, free storage of the object, perform save and restore operations for the object, or transfer ownership of the object. If a user has special save system authority (*SAVSYS), object existence authority is not needed. Object existence authority is required to create an object that has been named by an authority holder.

***OBJMGT:** Object management authority provides the authority to specify the security for the object, move or rename the object, and add members to database files.

***OBJOPR:** Object operational authority provides authority to look at the description of an object and use the object as determined by the user's data authorities to that object.

***OBJREF:** Object reference authority provides the authority needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user has this authority on a physical file, the user can add referential constraints in which the physical file is the parent. This authority is currently only used for database files.

***ADD:** Add authority provides the authority to add entries to an object (for example, job entries to a queue or records to a file).

***DLT:** Delete authority allows the user to remove entries from an object, for example, remove messages from a message queue or records from a file.

***EXECUTE:** Execute authority provides the authority needed to run a program or to locate an object in a library.

***READ:** Read authority provides the authority needed to show the contents of an entry in the object or to run a program.

***UPD:** Update authority provides the authority needed to change the entries in the object.

Example for RVKWSOAUT

```
RVKWSOAUT  WSOTYPE(*SGNUSL)  USER(HEANDERSON)
AUT(*DLT *UPD)
```

This command removes the delete and the update authorities for signed-on user list objects from the user profile name HEANDERSON.

No error messages.

ROLLBACK (Rollback) Command Description

ROLLBACK Command syntax diagram

Purpose

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

When the ROLLBACK command is issued:

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

More information on the use of commitment control is in the Backup, Recovery, and Availability topic in the Information Center.

There are no parameters for this command.

Example for ROLLBACK

```
ROLLBACK
```

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

Error messages for ROLLBACK

*ESCAPE Messages

CPF83D0

Commitment operation not allowed.

CPF835F

Commit or rollback operation failed.

CPF8350

Commitment definition not found.

CPF8359

Rollback operation failed.

CPF8367

Cannot perform commitment control operation.

*STATUS Messages

CPF83E6

Commitment control operation completed with resynchronization in progress.

RUNBCKUP (Run Backup) Command Description

RUNBCKUP Command syntax diagram

Purpose

The Run Backup (RUNBCKUP) command allows the user to run a predefined backup of specified objects to tape. The backup may include libraries (all user libraries or those selected in the backup list), folders (all folders or root folders selected in the backup list), security data, configuration data, mail, and calendars.

Required Parameter

BCKUPOPT

Specifies the backup options to use.

***DAILY:** The daily backup options are used.

***WEEKLY:** The weekly backup options are used.

***MONTHLY:** The monthly backup options are used.

Optional Parameter

DEV Specifies a list of tape devices to use for the backup.

***BCKUPOPT:** The tape device names stored in the specified options are used for the backup.

tape-device-name: Specify a list of tape devices used for the backup.

Examples for RUNBCKUP

Example 1: Running a Daily Backup

```
RUNBCKUP BCKUPOPT(*DAILY)
```

This command runs the daily backup using the devices specified in the options.

Example 2: Running a Monthly Backup

```
RUNBCKUP BCKUPOPT(*MONTHLY) DEV(TAP02)
```

This command runs the monthly backup using device TAP02 instead of those specified in the options.

Error messages for RUNBCKUP

*ESCAPE Messages

CPF1EE3

Not authorized to backup options.

CPF1EE4

Not authorized to run backup.

CPF1EE6

Devices specified cannot be used.

CPF1EE7

Unexpected error occurred during backup.

CPF1EE8

Unexpected error from exit program &2/&1.

CPF1E6C

Backup options in use.

CPF1E6E

Nothing selected for backup.

CPF1E63

Error occurred while checking backup authority.

CPF1E67

Backup options and library backup list damaged.

CPF1E68

Backup incomplete.

CPF1E99

Unexpected error occurred.



Run Java (RUNJVA) Command Description

Run Java Command syntax diagram

Purpose

The Run Java (RUNJVA) command runs the iSeries 400 Java program associated with the Java class specified. If no Java program exists, one will be created and associated with the class file.

This command can operate on files in any file system that supports the Integrated File System APIs.

Required Parameter

CLASS

Specifies the class name to be run. The class name may be qualified by one or more package names. Each package name must be followed by a period. For example, CLASS('pkg1.pkg2.myclass') would identify a class qualified by two package names.

class-name: Specify the name of the class to be run.

***VERSION:** The build version information for the Java Development Kit (JDK) and the Java Virtual Machine (JVM) is displayed. No Java program is run.

Optional Parameters

PARM Specifies one or more parameter values that are passed to the Java program. A maximum of 200 parameter values can be passed.

***NONE:** There are no input parameters to the Java program.

parameter-value: Specify the parameter value to be passed to the Java program.

CLASSPATH

Specifies the path used to locate classes. Directories are separated by colons.

***ENVVAR:** The class path is determined by the environment variable CLASSPATH.

class-path: Path used to locate classes. A example class path is '/directory1/directory2:/QIBM/ProdData/Java400'.

CHKPATH

Specifies the level of warnings given for directories in the CLASSPATH that have public write authority. A directory in the CLASSPATH that has public write authority may result in a security exposure where an undesired class is inserted into the directory to be found before the official class.

***WARN:** A warning message is sent for each directory in the CLASSPATH that has public write authority.

***SECURE:** A warning message is sent for each directory in the CLASSPATH that has public write authority. If one or more warning messages are sent, an escape message is sent and the Java program is not run.

***IGNORE:** Ignore the fact that directories in the CLASSPATH may have public write authority. No warnings messages are sent.

OPTIMIZE

Specifies how to treat class files if no Java program is associated with the file.

For 10,20,30,40 this specifies the optimization level of the iSeries 400 Java program which will be created if no Java program is associated with the Java class file. The Java program will contain machine instruction sequences that are run when the Java program is invoked and will remain associated with the class file after the Java program has been run.

If the Java class file is determined to be downlevel or out of date, a new Java program will be created using the optimization level that was used when the previous Java program was created, instead of the value specified for this parameter.

For OPTIMIZE(*INTERPRET), the resulting Java program will interpret the class file byte codes when invoked. OPTIMIZE(*INTERPRET) Java programs will be smaller but will run slower than Java programs created with higher optimization levels. As you increase the optimization level beyond 10, the Java program performance will generally improve, but the time required to create the Java program will increase and you will have less ability to debug the Java program.

For OPTIMIZE(*JIT), no Java program containing machine instruction sequences is created if no program is associated with the class file. Rather, the class is run using the Just In Time compiler (JIT).

10: The Java program will contain a compiled version of the class file byte codes but will have only minimal additional compiler optimization. Variables can be displayed and modified while debugging.

***INTERPRET:** The Java program created will not be optimized. When invoked, the Java program will interpret the class file byte codes. Variables can be displayed and modified while debugging.

***JIT:** No Java program containing machine instruction sequences is created. The class is run using the Just In Time compiler (JIT).

20: The Java program will contain a compiled version of the class file byte codes and will have some additional compiler optimization performed. Variables can be displayed but not modified while debugging.

30: The Java program will contain a compiled version of the class file byte codes and will have more compiler optimization performed than optimization level 20. During a debug session, user variables cannot be changed, but can be displayed. The presented values may not be the current value of the variable.

40: The Java program will contain a compiled version of the class file byte codes and will have more compiler optimization performed than optimization level 30. In addition, it includes optimization that disables call and instruction tracing.

INTERPRET

Specifies the mode in which to run the Java classes.

***OPTIMIZE:** Whether all Java class files are run interpretively depends on the value specified for the OPTIMIZE parameter. If OPTIMIZE(*INTERPRET) was specified, all Java class files will be run interpretively. If any other value was specified for the OPTIMIZE parameter, only Java class files with Java programs created using CRTJVAPGM command and specifying OPTIMIZE(*INTERPRET) will be run interpretively.

***NO:** Only Java class files with Java programs created using CRTJVAPGM command and specifying OPTIMIZE(*INTERPRET) will be run interpretively. Java class files that need a Java program created will be created with the optimization level specified in the OPTIMIZE parameter. This option allows a user to run attached DE programs, but creates *INTERPRET programs for those classes which do not already have existing programs attached by setting the OPTIMIZE(*INTERPRET).

***YES:** All Java class files will be run interpretively regardless of what OPTIMIZE value the associated Java program was created. Java class files that need a Java program created will use the optimization level specified in the OPTIMIZE parameter.

***JIT:** All Java class files will be run using the Just In Time compiler (JIT), regardless of what OPTIMIZE value was used when the associated Java program was created.

PROP Specifies a list of values to assign to Java properties. Up to 100 properties can have a value assigned.

***NONE:** No properties are specified.

Element 1: Property name

property-name: Specify the name of the property to be changed.

Element 2: Property value

property-value: Specify the value to be assigned to the property.

GCHMAX

Specifies the maximum size, in kilobytes, that the garbage collection heap can grow to. This is used to prevent runaway programs that consume all of the available storage. Normally, garbage collection runs as an asynchronous thread in parallel with other threads. If the maximum size is reached, all other threads are stopped while garbage collection takes place. This may adversely affect performance.

***NOMAX:** The heap will grow until all system resources are depleted. Then a synchronous garbage collection is started to reclaim resources no longer in use.

heap-maximum-size: The value in kilobytes that the garbage collection heap can grow to. It is not recommended to set the maximum size less than 32768 kilobytes.

GCHINL

Specifies the initial size, in kilobytes, of the garbage collection heap. This is used to prevent garbage collection from starting on small programs.

2048: The default initial size is 2048 kilobytes.

heap-initial-size: The initial value of the garbage collection heap in kilobytes. It is not recommended to set the initial size less than the default of 2048 kilobytes.

GCFRQ

Specifies the relative frequency that garbage collection runs.

This parameter is no longer supported. It exists solely for compatibility with releases earlier than Version 4 Release 3 Modification 0 of the AS/400 system.

GCPTY

Specifies the priority of the tasks running garbage collection.

This parameter is no longer supported. It exists solely for compatibility with releases earlier than Version 4 Release 3 Modification 0 of the AS/400 system.

OPTION

Specifies special options used when running the Java class.

***NONE:** No special options will be used when running the Java class.

***DEBUG:** Allows the iSeries 400 system debugger to be used for this Java program.

***VERBOSE:** A message is displayed each time a class file is loaded.

***VERBOSEGC:** A message is displayed for each garbage collection sweep.

***NOCLASSGC:** Unused classes are not reclaimed when garbage collection is run.

OUTPUT

Specifies where output from the Java program should be sent and, if output is directed to the Java shell display, whether the shell display panel should go away when the Java program ends.

Element 1:Destination

*****: A Java shell display panel will be used to display output if the Java program is run from an interactive job. If the Java program is run in a batch job, the Java program output is sent to a spooled file through the QPRINT printer device file.

***NONE:** The Java program output is discarded.

***PRINT:** The Java program output is sent to a spooled file through the QPRINT printer device file.

Element 2:Program End Action ***PAUSE:** The Java shell display panel stays until the F3, F12, or Enter key is pressed.

***CONTINUE:** The Java shell display panel goes away when the Java program ends.

Example for RUNJVA

Example 1: Run a Java Program

```
RUNJVA CLASS('projectA.myJavaclassname')
```

This command will run the iSeries 400 Java program associated with the class myJavaclassname.

Error messages for RUNJVA

*ESCAPE Messages

JVAB534

Unable to complete Java program "&1".

JVAB535

Unmonitored exception received.

JVAB537

Java shell already active in job.

JVAB538

Error occurred when running Java shell.

JVAB539

Unable to start system debugger.

JVAB53A

Unable to start Java shell, reason code &1.

JVAB53B

Java processing canceled.

JVAB53D

Java Development Kit could not be found.

JVAB546

Error detected while running java in batch mode.



RUNLPDA (Run LPDA-2) Command Description

RUNLPDA Command syntax diagram

Purpose

The Run LPDA-2 (RUNLPDA) command allows you to run a Link Problem Determination Aid-2 (LPDA-2) operational command on local or remote data circuit-terminating equipment (DCE). The RUNLPDA command can be used to:

- Establish or disconnect a switched telephone network connection.
- Open or close the relay contact in a coupler.
- Determine whether a relay contact is open or closed.
- Determine whether electric current is flowing through an internal sensor.
- Change the transmit speed of a DCE to full or backup.

The result of the RUNLPDA command is returned as a message.

Restrictions:

1. The RUNLPDA command is valid only for an analog LPDA-2 DCE attached to a nonswitched SDLC line.
2. This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.

Required Parameters

LINE Specifies the name of the nonswitched SDLC line that is attached to the analog DCE on which the LPDA-2 operational command is to be run. If *CALL is specified on the OPTION parameter, the line must be varied on but not active. If *CONTOPER, *CONTSENSE, *DSC, or *SETSPEED is specified on the OPTION parameter, the line must be either varied on or active.

OPTION

Specifies which LPDA-2 operational command is run.

The contact operate, contact sense, and set transmit speed commands can be run on a local DCE or a remote DCE.

To run one of these commands on a local DCE:

- On the LCLDCEADR parameter, specify the address of the local DCE.
- On the RMTDCEADR parameter, specify *NONE.

To run one of these commands on a remote DCE:

- On the LCLDCEADR parameter, specify the address of the local DCE to which the remote DCE is connected.
- On the RMTDCEADR parameter, specify the address of the remote DCE.

Note:

If the local DCE is configured as point-to-point secondary or multipoint tributary, LPDA-2 commands are not sent to the remote DCE.

***CALL:** The call out command is run. This command establishes a connection between a local and a remote DCE over a switched telephone network. This value is valid only if:

- Both the local DCE and the remote DCE have two-wire couplers installed or both the local DCE and the remote DCE have four-wire couplers installed.
- The line specified on the LINE parameter is varied on, but not active.

***CONTOPER:** The contact operate command is run. This command opens or closes the relay contact in the coupler, depending on the value specified on the CONTACT parameter. This value is valid only if a two-wire coupler is installed in the DCE on which this LPDA-2 command is run.

***CONTSENSE:** The contact sense command is run. This command reports whether the relay contact in the coupler is open or closed and whether electric current is flowing through the internal sensor. This option is valid only if a two-wire coupler is installed in the DCE on which this command is run.

***DSC:** The disconnect command is run. This command disconnects the switched telephone network connection between the local DCE and the remote DCE.

***SETSPEED:** The set transmit speed command is run. This command changes the transmit speed of the DCE to full or backup, depending on the value specified on the SPEED parameter. For multipoint DCE configurations in which the data terminal equipment (DTE) ports can be set to different speeds, use the DTEPORT parameter to specify the port.

Note:

The set transmit speed command may not change the transmit speed, depending on the configuration options selected for the DCE. Refer to the DCE documentation for more information.

Optional Parameters**LCLDCEADR**

Specifies the hexadecimal address of the local DCE. Refer to the DCE documentation for more information on addressing.

***LCL:** X'01' is used for the address.

local-DCE-address: Specify the address of the local DCE. Valid values range from X'01' through X'FB'.

RMTDCEADR

Specifies the hexadecimal address of the remote DCE on which the LPDA-2 operational command is to be run.

***NONE:** X'00' is used as the address, which indicates that the LPDA-2 command is to be run on the local DCE.

***ANY:** X'FD' is used for the address. Specify this value in the following situations:

- The LPDA-2 command is to be run on any remote DCE connected to the local DCE.
- You do not know the remote DCE address on a point-to-point line.
- To run the LPDA-2 operational command on all tributary DCEs on a multipoint line. In this case, no detailed response is received.

remote-DCE-address: Specify the address of the remote DCE. Valid values range from X'01' through X'FB'.

TELNBR

Specifies the telephone number or telephone numbers that the local DCE dials to establish a connection to the remote DCE.

***STORED:** The telephone number or numbers stored in the local DCE when the DCE was configured are used.

telephone-number: Specify one telephone number if a two-wire coupler is installed in the local DCE. Specify two telephone numbers if a four-wire coupler is installed in the local DCE. Only numeric characters are processed by the LPDA-2 command, but you can also enter alphabetic characters or any other non-DBCS characters to improve readability. A comma (,) can be used to instruct the DCE to pause during dialing.

CONTACT

Specifies whether to open or close the relay contact in a two-wire coupler.

***OPEN:** The relay contact is opened.

***CLOSE:** The relay contact is closed.

SPEED

Specifies the desired transmit speed of the DCE.

***FULL:** The transmit speed is set to full.

***BACKUP:** The transmit speed is set to backup.

DTEPORT

Specifies the DTE port on the local or remote DCE for which the transmit speed is changed. This parameter is applicable only to multiport DCEs that do not use the multi-address configuration option.

***ALL:** The aggregate speed of the DCE is changed. Refer to the DCE documentation for information on the effect of changing the aggregate speed on the transmit speed of individual ports.

A: The transmit speed of the A-port is changed.

B: The transmit speed of the B-port is changed.

C: The transmit speed of the C-port is changed.

D: The transmit speed of the D-port is changed.

DCERTY

Specifies whether the local DCE resends the LPDA-2 command to the remote DCE if no response is received from the remote DCE. No retry can be attempted if OPTION(*CALL) or OPTION(*DSC) is specified.

***NO:** No retry is attempted.

***YES:** One retry is attempted.

Examples for RUNLPDA

Example 1: Establishing a Switched Telephone Network Connection

```
RUNLPDA LINE(SDLCLINE) OPTION(*CALL)
LCLDCEADR(*LCL) RMTDCEADR(*NONE)
TELNBR(*STORED)
```

This command runs the call out command. The local DCE with address X'01' (*LCL) on line SDLCLINE dials the telephone numbers that are stored in the local DCE.

Example 2: Establishing a Switched Telephone Network Connection

```
RUNLPDA LINE(SDLCLINE) OPTION(*CALL)
LCLDCEADR(*LCL) RMTDCEADR(*NONE)
TELNBR('9, 1-507-555-1212' '9, 1 (507) 555-1313')
```

This command runs the call out command. The local DCE dials the two numbers specified on the TELNBR parameter. The comma (,) indicates a pause during dialing. Other non-numeric characters are ignored, but are allowed for easier reading.

Example 3: Disconnecting a Switched Telephone Network Connection

```
RUNLPDA LINE(SDLCLINE) OPTION(*DSC)
LCLDCEADR(10) RMTDCEADR(*ANY)
```

This command runs the disconnect command. The local DCE with address X'10' disconnects from the switched telephone network.

Example 4: Closing the Relay Contact in the Local DCE

```
RUNLPDA LINE(SDLCLINE) OPTION(*CONTOPER)
LCLDCEADR(02) RMTDCEADR(*NONE)
CONTACT(*CLOSE)
```

This command runs the contact operate command. The local DCE with address X'02' closes the relay contact in its two-wire coupler.

Example 5: Reporting the Status of the Relay Contact

```
RUNLPDA LINE(SDLCLINE) OPTION(*CONTSENSE)
LCLDCEADR(01) RMTDCEADR(04)
```

This command runs the contact sense command. A message reports the status of the relay contact in the remote DCE with address X'04'. (The correct local DCE address must be specified on the LCLDCEADR parameter.)

Example 6: Changing the Transmit Speed

```
RUNLPDA LINE(SDLCLINE) OPTION(*SETSPEED)
LCLDCEADR(05) RMTDCEADR(*NONE)
SPEED(*BACKUP) DTEPORT(B)
```

This command runs the set transmit speed command. The transmit speed for Port B of the local DCE with address X'05' is changed to backup speed.

Error messages for RUNLPDA

*ESCAPE Messages

CPF1BAA

LPDA-2 command cannot run in switched network backup.

CPF1BAB

LPDA-2 command not supported by target DCE.

CPF1BAC

Required feature not installed.

CPF1BAD

LPDA-2 command not compatible with DCE configuration.

CPF1BA1

Line description &1 does not exist.

CPF1BA2

Line &1 not active or not varied on.

CPF1BA3

Line &1 is active secondary.

CPF1BA4

Line &1 failed while processing RUNLPDA.

CPF1BA5

No response received for LPDA-2 command.

CPF1BA6

Cannot run LPDA-2 command on line &1.

CPF1BA7

No buffers available for RUNLPDA data.

CPF1BA8

LPDA-2 command already active on line &1.

CPF1BBA

Continuous answer tone received for first call.

CPF1BBB

Continuous answer tone received for second call.

CPF1BBC

No line signal on switched connection.

CPF1BBD
No initial dial tone.

CPF1BBE
No dial tone after pause.

CPF1BBF
Busy tone received from remote DCE.

CPF1BB0
Coupler not installed in local DCE.

CPF1BB1
Coupler not operational.

CPF1BB2
LPDA-2 command not valid with configuration options.

CPF1BB3
Telephone number not valid.

CPF1BB5
Switched connection already active.

CPF1BB6
Only single telephone number allowed.

CPF1BB7
Two telephone numbers required.

CPF1BB8
No answer from remote DCE on first call.

CPF1BB9
No answer from remote DCE on second call.

CPF1BDA
Error occurred processing RUNLPDA command.

CPF1BDB
Internal processing error.

CPF1BDC
Required feature not operational.

CPF1BDD
Ring back limit exceeded.

CPF1BDE
DCE busy. LPDA-2 command cannot be run.

CPF1BDF
Line not in proper state for LPDA-2 call out command.

CPF1BD3
DCE currently processing previous LPDA-2 command.

CPF1BD5
Cannot send command to remote DCE.

CPF1BD6
Line description &1 not nonswitched *SDLC.

CPF1BD8
Request failed. System service tools active.

CPF1BD9

RUNLPDA command does not support switched lines.

CPF1B85

Two-wire coupler not installed.

CPF1B86

Coupler not installed.

CPF1B87

Transmit speed cannot be changed.

CPF1B88

DTE port selected not valid for configuration.

CPF1B94

Local DCE received no response from remote DCE.

CPF1B95

Not authorized to line description &1.

CPF1B97

Format of response received not valid.

CPF1B98

Received response with bad frame check sequence.

OFC8DB7

Internal processing failure.

RUNQRY (Run Query) Command Description

RUNQRY Command syntax diagram

Purpose

The Run Query (RUNQRY) command runs an existing query or a default query if only a file name is specified for this command. The query gets information from the system database and produces a report of that information. The report is created in either detailed or summary form. The definition of the query can be printed when output to a printer or database file is specified. The output is shown, printed, or stored in a database file.

The command is used in three ways: to run an existing query (one that has already been created), to run an existing query with some of its values changed by values specified on this command, or to run a default query based only on the defaults and values specified in this command.

- To run an existing query without changing the file or files to query, use the QRY parameter (without the QRYFILE parameter) to specify the name of the query.
- To run a changed version of an existing query, use the QRY parameter and the appropriate parameters to change the definition as desired. The parameter values specified on this command override the corresponding values in the existing query definition, but only when the command is processing. For example, use the QRYFILE parameter to indicate a different file or list of files to use in the query.
- To query a file without a previously defined query definition, use the QRYFILE parameter to specify which file to query. Only one file name can be specified for a default query.

If the user specifies both the QRY and QRYFILE parameters, the files specified in the QRYFILE parameter override the file names specified in the query. Therefore, if multiple files (and members) are defined in the query definition and the user wants to change one or two of them, specify *SAME for the file selections that do not change, and specify the values for the files to override.

Notes:

1. When a changed version of an existing query is run, the changes specified on the RUNQRY command do *not* change any of the values in the query definition itself; they affect only the results of the report being run.
2. If an existing query (identified on the QRY parameter) is used, *RUNOPT is the default value for any unspecified parameters; that is, the same value specified (or assumed) in the definition of the query is used as the default. If this is a default query, the default value is not *RUNOPT, but is the next predefined value listed in the syntax diagram.

Required Parameters

QRY Specifies the name of an existing query to run. If QRY is not specified, QRYFILE must be specified.

***NONE:** No existing query definition is used. Instead, a default query (or quick query) is used to get information from the file specified on the QRYFILE parameter.

If no value is specified, the file or files that were specified when the query was defined are the files to be queried.

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

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

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

library-name: Specify the name of the library to be searched.

query-name: Specify the name of the query to run.

QRYFILE

Specifies the database file or files to be queried for information. If the QRY parameter is specified, as many as 32 files can be specified on this parameter by using the file names and/or using the default value *SAME for one or more of the files. If the QRY parameter is not specified, only one file name can be specified on this parameter. If QRYFILE is not specified, QRY must be specified.

If no value is specified, the file or files that were specified when the query was defined are the files used to run the query.

***SAME:** The list of files remains the same as defined in the query definition.

Element 1: Database File Name

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

***SAME:** The value of the library, file, and member do not change for the file in this position within the list. The values are the same as defined in the query. Any values specified for the library, file, and member are ignored.

***LIBL:** All libraries in the 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.

***RUNOPT:** The library specified in the query definition for the file selection is used.

library-name: Specify the name of the library to be searched.

database-file-name: Specify the names of one or more database files that contain the data from which the system gets information to produce the output. Up to 32 files can be specified.

Element 2: Database File Member

***FIRST**: The first member in the file is the member to query.

***RUNOPT**: The member specified in the query definition for this file selection is used.

***LAST**: The last member in the file is the member to query.

member-name: Specify the name of the file member to query.

Optional Parameters

OUTTYPE

Specifies where the report or output produced by the query is sent. If a value is not specified in the query and is not entered on the command, or if a query name is not specified, *DISPLAY is assumed.

***RUNOPT**: If a query definition is being used, the type of output specified in the query definition is the type produced when this query is run.

***DISPLAY**: The output produced by the query is sent to the display station that runs the command. If the command is being run in batch, the output is sent to the printer instead of the display.

***PRINTER**: The output produced by the query is printed.

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

OUTFORM

Specifies the form of output produced by the query. If no value was specified in the query, and none was entered on the command, or if a query name is not specified, *DETAIL is assumed.

***RUNOPT**: If a query definition is used, the output form specified in the query definition is used when this query is run.

***DETAIL**: The output form produced by the query is a report containing detail records and summary records if any exist.

***SUMMARY**: The output form produced by the query is a report containing summary records only.

RCDSLTL

Specifies whether the query is run with a run time selection test.

Note:

The iSeries 400 Query/400 licensed program must be installed and the query must be run interactively to specify *YES.

***NO**: The query is run without showing the record selection display in Query.

***YES**: Record selection definition is allowed for this run only. A display is shown on which the user can change the record selection tests defined in the query or specify record selection tests if a query name was not specified.

Note: For output to display, query can be run repeatedly.

PRTDFN

Specifies whether the query definition is printed with the report when the query is run. The definition can be printed when the output of the query is printed or stored in a data base file, as determined by the OUTTYPE parameter. If a value is not specified in the query or in this parameter, or if a query name is not specified, the value *NO is assumed.

***RUNOPT:** If a query definition is used when the query is run, the print option specified in the query definition is used.

***NO:** The query definition is not printed when the query is run.

***YES:** The query definition is printed in the report. *YES cannot be specified if OUTTYPE(*DISPLAY) is specified or assumed.

OUTFILE

Specifies the database file (if any) that receives the query output. If no value is specified for this parameter, the library, file, member and option specified in the query are assumed. If a query is not specified, the file QQRVOUT is created in the default output file library. This library is defined in the query profile or the current library (*CURLIB) if no profile exists. The first member (*FIRST) of this new file is used for the output.

Note: If the user did not specify a current library, the QGPL library is used.

If the name specified by the OUTFILE parameter does not exist, the system creates it in the specified library.

***RUNOPT:** The database file specified in the query is used to receive the output of the query. The file, library, member and option do not change. Any values specified for the library, member, and option are ignored.

Element 1: Database File Name

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

***RUNOPT:** If specified in the query definition, the output is directed to the library named in the query definition.

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

library-name: Specify the name of the library to be used.

database-file-name: Specify the name of the database file that receives the output of the command.

Element 2: Database File Member

***FIRST:** The first member in the file receives the query output.

***LAST:** The last member in the file receives the query output.

***RUNOPT:** The member specified in the query receives the query output.

member-name: Specify the name of the file member used to receive the query output.

Element 3: Adding or Replacing Data

The last option specifies whether to put the data in a new database file, replace an existing database file, add a new member, replace an existing member, or add data to an existing member. If no value is specified in the query or in this parameter, or if a query name is not specified, the value *NEWFILE is assumed.

***RUNOPT:** If a query definition is used, the member option specified in the query definition is the type used when this query is run.

***NEWFILE:** The output is written to a new database file.

***RPLFILE:** The output deletes the old file and creates a new file.

***NEWMBR:** The output is added as a new member.

***RPLMBR:** The existing member is cleared and the output is then added.

***ADDMBR:** The output is added to the end of an existing member.

AUT Specifies the authority given to users who do not have specific authority to the output file, who are not on an authorization list, and whose user group has no specific authority to the output file. More information on this parameter is in Commonly used parameters.

Note: The authority is assigned only when the output file is created or replaced.

***RUNOPT:** If specified in the query definition, the authority named in the query definition is used.

***LIBCRTAUT:** The public authority for the output file is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the output file). The public authority is determined when the output file is created. If the CRTAUT value for the library changes after the output file is created, the new value does not affect any existing objects.

***CHANGE:** The user can perform all operations on the output file except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the output file. Change authority provides object operational authority and all data authority.

***USE:** The user can perform basic operations on the output file, such as running a program or reading a file. The user cannot change the output file. *USE authority provides object operational authority, read authority, and execute authority.

***ALL:** The user performs all operations on the output file except those limited to the owner or controlled by authorization list management authority.

***EXCLUDE:** The user cannot access the output file.

authorization-list-name: Specify the name of the authorization list used.

PRTDEV

Specifies the printer device on which the report is printed. If no value is specified, the printer that was specified when the query was defined is assumed. If no printer is specified in the query or in this parameter, or if a query name is not specified, the value *PRINT is assumed.

Note:

If an override is in effect for the printer file QPQUPRFIL, this parameter uses the value specified by the override.

***RUNOPT:** If a query definition is used, the printer specified in the query definition is used to print the output when this query is run.

***PRINT:** The default printer, as defined by QPQUPRFIL, is used to print the output when this query is run.

printer-device-name: Specify the name of the printer that is used to print the output when this query is run.

FORMSIZE

Specifies the length and the width of the forms on which the report is printed. If a form width greater than 132 is specified, Query will open the file with CPI(15) and FONT(*CPI). If no value is specified in the query or in this parameter, or if a query name is not specified, 132 is the assumed form width, and the value from the file QPQUPRFIL is the assumed form length.

Note:

If an override is in effect for the printer file QPQUPRFIL, this parameter uses the value specified by the override.

Element 1: Form Length

***RUNOPT:** If a query definition is being used when the query is run, the form size specified in the query definition is used. If the form size specified in the query definition is blank, the value from QPQUPRFIL is assumed.

form-length: Specify the form length used when this query is run. Valid values range from 1 through 255.

Element 2: Form Width

***RUNOPT:** If a query definition is being used when the query is run, the form size specified in the query definition is used. If the form size specified in the query definition is blank, the value from QPQUPRFIL is assumed.

form-width: Specify the form width used when this query is run. Valid values range from 1 through 378.

FORMTYPE

Specifies the type of form on which the output is printed. The identifiers used to indicate the type of forms are user-defined and can be a maximum of 10 characters in length.

Note:

If a value is not specified in the query or on this parameter, or if a query name is not specified, the value in QPQUPRFL is assumed. If an override is in effect for the printer file QPQUPRFL, this parameter uses the value specified by the override.

***RUNOPT:** If a query definition is used, the form type specified in the query definition is used when this query is run.

form-type: Specify the form type that is used when this query is run.

COPIES

Specifies the number of copies being printed.

Note:

If a value is not specified in the query or on this parameter, or if a query name is not specified, 1 is the assumed number of copies. If an override is in effect for the printer file QPQUPRFL, this parameter uses the value specified by the override.

***RUNOPT:** If a query definition is used, the number of copies specified in the query definition is used when this query is run. If the number of copies specified in the query definition is blank, the number of copies from QPQUPRFL is assumed.

number-of-copies: Specify the number of copies to print when this query is run. Specify a number ranging from 1 through 255.

LINESPACE

Specifies the number of blank lines to leave between lines in the report. The numbers range from 1 to 3. If a value was not specified in the query or on this parameter, or if a query name is not specified, 1 is the assumed value.

***RUNOPT:** If a query definition is used, the number of lines specified in the query definition is used when this query is run.

1: Indicates that single spacing (no blank lines) is used when the query output is printed.

2: Indicates that double spacing (1 blank line) is used when the query output is printed.

3: Indicates that triple spacing (2 blank lines) is used when the query output is printed.

Examples for RUNQRY**Example 1: Printing Summary Records Only**

```
RUNQRY QRY(LIBX/QRY1) OUTTYPE(*PRINTER)
      OUTFORM(*SUMMARY) COPIES(4)
```

This command runs the query QRY1 located in library LIBX. The report that is produced and printed contains summary records only. Four copies of the report are printed.

Example 2: Running a Default Query

```
RUNQRY QRYFILE((LIBX/FILE2 *FIRST)) OUTTYPE(*OUTFILE)
      OUTFORM(*DETAIL) RCDSL(*YES)
      OUTFILE(LIB2/OUT1 MBR4 *NEWMBR)
```


This command runs a default query and gets the data from the first member of file FILE2 located in library LIBX. Member MBR4 is created as a new member to file OUT1 in library LIB2. Member MBR4 contains the output from the default query. The record selection display is shown to allow you to specify which records from file FILE2 in library LIBX are written to new member MBR4 in file OUT1 in library LIB2. The output contains detail records only.

Error messages for RUNQRY

None

AREXEC (Run Remote Command) Command Description

AREXEC Command syntax diagram

For the description of the AREXEC command, see the RUNRMTCMD (Run Remote Command) command description.

RUNRMTCMD (Run Remote Command) Command Description

RUNRMTCMD Command syntax diagram

Purpose

The Run Remote Command (RUNRMTCMD) command, also known as AREXEC when an SNA address is specified for the remote location name, allows iSeries 400 users to run a command on a remote system that is running the target portion of this function.

The target portion of this function can be a *rexecd* (remote executing) daemon if you specify *IP for the address type, or an *AREXECD* (APPC remote executing) daemon when an SNA address is specified for the remote location name. If you want to run a command on a Client Access/400 client that supports this function, you will need to specify an SNA address for the remote location name.

When the command is sent to the remote system, the local system waits for the command to complete and the output from the remote command will be sent back to the local system and placed in a spooled file.

Required Parameters

CMD Specifies a character string of up to 2000 characters that represents a command that is run on the target system. The maximum length supported by the target system may be less than 2000 characters. If you specify a command string that exceeds the maximum length supported by the target system, the command will fail.

The command must be enclosed in apostrophes if it contains embedded blanks or special characters.

Note:

The normal rule of pairing apostrophes in quoted strings on the local system must be *doubled* when the same string is submitted to a remote system on this CMD parameter; this is required because the user is coding a quoted string within another quoted string. Therefore, when this parameter is being coded, wherever a single apostrophe would normally be paired with another apostrophe, *each occurrence in the inside set of apostrophes must be doubled* to produce the same results at the target system.

RMTLOCNAME

If using the AREXEC command, RMTLOCNAME must be a single-element value containing an SNA address. The second element value *SNA is implied and cannot be specified.

Specifies the name of the target system and the type of address used to connect to the target system.

Element 1: Name or address

When the address type is *SNA, specify the name of the target system using the format nnnnnnnn.cccccc, where nnnnnnnn is the network ID and cccccc is the remote location name. If no network ID is specified, the command uses the network attributes to determine the default network ID.

When the address type is *IP, specify a host name or an Internet address to identify the target system. Host names must follow these conventions:

- The first character must be either A through Z or 0 through 9.
- The last character must be either A through Z 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.

The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Element 2: Type

Specifies the address type for the target system that the command is to be run on.

***SNA:** The target system is accessed using a Systems Network Architecture (SNA) address and protocol.

***IP:** The target system is found using a host name or an Internet address over a TCP/IP connection.

Optional Parameters

RMTUSER

Specifies the user identifier (ID) for the target system. If a user ID is specified for this parameter and password security is active on the target system, RMPWD(*NONE) is not valid.

***NONE:** No user ID is sent. If security on the target system is configured to require a user ID, the command will fail.

***CURRENT:** The user ID of the job (signed-on user) using this command is sent.

remote-user-identifier: Specify a user ID to use that exists on the target system. If a user ID is specified and password security is active on the target system, a password must be specified, even if the profile specified is the same as the current profile.

RMPWD

Specifies the password sent to the target system.

***NONE:** The system does not pass a password. If a user identifier (ID) is specified on the RMTUSER parameter and password security is active on the target system, the command will fail.

password: Specify a password sent to the target system to verify the sign-on of the user ID specified in the RMTUSER parameter. If a address type of *IP is specified, the password sent is not substituted across the communication line; if *SNA is specified, the password may or may not be substituted, depending on whether the remote system supports password substitution.

MODE Specifies the mode used when sending the command to the target system. This parameter is allowed only when the address type value is *SNA.

***NETATR**: The mode in the network attributes is used.

mode-name: Specify a mode name. Specify BLANK for a mode name consisting of eight blank characters.

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

CCSID

Specifies the coded character set identifier (CCSID) that will be used for conversions. When data is sent to the remote system, the data is converted using the CCSID of the job to the CCSID specified on the CCSID parameter. Similarly, when data is received from the remote system, the data is converted from the CCSID that was specified on the CCSID parameter to the CCSID of the job.

Note:

If the CCSID of the job is 65535 (indicating no conversion), the default CCSID of the job will be used.

This command uses the round-trip conversion method when converting data from the source CCSID to the target CCSID and back. For more information about CCSID conversion methods, see the Globalization topic in the Information Center.

***CALC**: The CCSID value sent from the target system is used if it is available. If a value is not provided from the target system, a default value of 00819 (ISO 8859-1 8-bit ASCII) is used.

coded-character-set-identifier: The requested CCSID value is used. This value is validated to ensure a valid CCSID has been requested.

WAITTIME

Specifies the time in seconds to wait for the response before declaring the remote location to be unreachable. This parameter is allowed only when the address type value is *SNA.

***NOMAX**: The system waits indefinitely.

***NOWAIT**: The system returns immediately if there is not a connection ready and available.

wait-time: Specify the wait time in seconds. Valid values are 2 to 3600 seconds.

Examples for RUNRMTCMD

Example 1: Run Remote Command (RUNRMTCMD) using SNA address type

```
RUNRMTCMD CMD('dir') RMTLOCNAME((APPN.RMTSYS *SNA))
```

This command runs the 'dir' command on the system specified by 'RMTLOCNAME' using an SNA address type.

Example 2: Run Remote Command (AREXEC) using SNA address type

```
AREXEC CMD('dir') RMTLOCNAME(APPN.RMTSYS)
```

This command is equivalent to the command in example 1.

Example 3: Run Remote Command Using IP Address Type and Host Name to Specify Target System

```
RUNRMTCMD CMD('ls') RMTLOCNAME((MYSYS.NET1.LOCAL *IP))
```

This command runs the 'ls' command on the system specified by host name MYSYS.NET1.LOCAL.

Example 4: Run Remote Command Using IP Address Type and Internet Address to Specify Target System

```
RUNRMTCMD CMD('ls') RMTLOCNAME(('9.5.1.94' *IP))
```

This command runs the 'ls' command on the system specified by Internet address '9.5.1.94'.

Example 5: Run Remote Command With Multiple Commands Using IP Address Type and Internet Address to Specify Target System

```
RUNRMTCMD CMD('ls; cat myfile; date')  
RMTLOCNAME(('9.5.1.94' *IP))
```

This command runs multiple commands, first 'ls', then 'cat myfile', then 'date' on system specified by Internet address '9.5.1.94'.

Example 6: Run Remote Command With a Wait Time Expiration Value

```
RUNRMTCMD CMD('dir') RMTLOCNAME((APPN.RMTSYS *SNA)) WAITTIME(15)
```

This command runs the 'dir' command on the specified system. The maximum wait time for the remote location to respond is 15 seconds. Only allowed using *SNA address type value.

Error messages for RUNRMTCMD

***ESCAPE Messages**

CPF91CB

Problems occurred on the command, but the command completed.

CPF91CC

Command did not complete successfully.

CPF91CF

Command failed on remote system.

CPF91C9

MODE value not allowed when *IP specified for RMTLOCNAME address type.



Printed in U.S.A.