IBM

IBM Systems - iSeries

i5/OS Commands
Starting with RSTSYSINF (Restore System Information)

Version 5 Release 4

IBM

IBM Systems - iSeries
i5/OS Commands
Starting with RSTSYSINF (Restore System Information)

Version 5 Release 4

IBM

Contents

Restore System Information (RSTSYSINF)

The Restore System Information (RSTSYSINF) command restores the subset of system data and objects saved by the Save System Information (SAVSYSINF) command.

RSTSYSINF is not to be used for system upgrades or migrations.

Security related system values may not be restored if they have been locked. For information on how to lock and unlock security related system values, see the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

The QPWDLVL (Password level) system value will not be restored. See Planning Password Level Changes in Chapter 7 of the iSeries Security Reference, SC41-5302 before changing the QPWDLVL system value. Also, based on the current setting of the QPWDLVL system value, QPWDMAXLEN (Maximum password length), QPWDMINLEN (Minimum password length), and QPWDVLDPGM (Password validation program) system values may not be restored.

Note: The RSTSYSINF command issues several restore commands for restoring objects. Parameters may or may not be used for all restore commands.

For more information, refer to the Backup and Recovery information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): *Name*	Required, Positional 1
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional
SEQNBR	Sequence number	1-16777215, ***SEARCH**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
SAVF	Save file	*Qualified object name*	Optional
	Qualifier 1: Save file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OPTFILE	Optical file	*Path name*, *****	Optional
OUTPUT	Output	***NONE**, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	*Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Keyword	Description	Choices	Notes
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	

Top

Device (DEV)

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

This is a required parameter.

Single values

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

Other values

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

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

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

Top

Volume identifier (VOL)

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

Single values

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

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

Other values (up to 75 repetitions)

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

Sequence number (SEQNBR)

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

*SEARCH
> The volume placed in a device is searched for a data file containing the saved system information objects. When a match is found, the system information objects are restored.

1-16777215
> Specify the sequence number of the file to be used for the restore operation.

End of media option (ENDOPT)

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

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

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

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

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

Save file (SAVF)

Specifies the save file used to restore the data.

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

Qualifier 1: Save file

name   Specify the name of save file to be used.

Qualifier 2: Library

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

*CURLIB
> The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

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

Optical file (OPTFILE)

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

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

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

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

'*optical-file-path-name*'
> Specify the path name of the optical file.

Output (OUTPUT)

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

*NONE
> No output is created.

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

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

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

File to receive output (OUTFILE)

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

Qualifier 1: File to receive output

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

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Output member options (OUTMBR)

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

Element 1: Member to receive output

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

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

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

Element 2: Replace or add records

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

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

Examples

Example 1: System Information Restored from TAP01
```
RSTSYSINF DEV(TAP01)
```

This command restores the system information from the tape put on the TAP01 tape drive.

Example 2: System Information Restored from SAVF and Printed Output Generated
```
RSTSYSINF DEV(*SAVF) SAVF(QGPL/SAVF) OUTPUT(*PRINT)
```

This command restores the system information from the save file named SAVF in library QGPL. Information about what was restored will be written to a spooled file.

Error messages

CPF38A8

     RSTSYSINF completed. One or more objects not restored.

Restore User Profiles (RSTUSRPRF)

The Restore User Profile (RSTUSRPRF) command restores the basic parts of a user profile or a set of user profiles that were saved by the Save System (SAVSYS) command or Save Security Data (SAVSECDTA) command. The Restore User Profile (RSTUSRPRF) command restores only the special authority granted in the Create User Profile (CRTUSRPRF) command; it does not restore the authority for the named objects owned by other users. To restore authority for objects owned by other users, the Restore Authority (RSTAUT) command must be used after the profiles, libraries, and objects are restored. If all user profiles are being restored, authorization lists and authority holders that existed when the SAVSYS or SAVSECDTA command was run are also restored.

If you specify USRPRF(*ALL) or SECDTA(*DCM), all other operations on the system must be stopped. This requires ending all subsystems through the End Subsystem (ENDSBS(*ALL)) command or End System (ENDSYS) command or specifying this command when the operating system is started. The RSTUSRPRF command is normally used after the restore of the operating system but before the user libraries are restored. The user profiles must be restored before any libraries or objects belonging to them can be restored. After the libraries and their objects are restored, the authority for the objects is restored to the user profiles by the RSTAUT command. At the completion of the command, either message CPF3775 or message CPC3705 is sent to QHST. More information on restoring the system is in the Backup and Recovery book, SC41-5304.

The following situations may apply to user profiles being restored by the RSTUSRPRF command:

- If a user profile exists on the system, but not on the media, the system profile remains.
- If a user profile exists on the media, but not on the system, a new user profile is created.
- If the user profile exists on both the media and the system, the media user profile is restored.
- If the user profile exists on the media and is being restored individually, the new user profile is created without its password or group connection.
- If the user profile exists on both the media and the system, and it is being restored individually, the media user profile is restored. However, the password and group connection on the system remains unchanged.
- If all user profiles are being restored, the passwords and group connections are also restored from the media.
- If user profiles exist on the system, there are no changes to the existing object authorities.

Note: This command ignores all file overrides that are currently in effect for the job.

Restrictions:

- You must have save system (SAVSYS) special authority to run this command.
- This command is shipped with no public authority (*EXCLUDE).
- If you specify USRPRF(*ALL) or SECDTA(*DCM), all other operations on the system must be ended. The End System (ENDSYS) or End Subsystem (ENDSBS) command can be used to end these operations. You must have job control (*JOBCTL) special authority to use the ENDSYS or ENDSBS command.
- You must specify USRPRF(*ALL) to restore authorization lists and authority holders.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): *Name*	Required, Positional 1
USRPRF	User profile	Single values: ***ALL**, *NEW, *NONE Other values (up to 300 repetitions): *Generic name, name*	Optional, Positional 2
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional, Positional 3
SEQNBR	Sequence number	1-16777215, ***SEARCH**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
SAVF	Save file	*Qualified object name*	Optional
	Qualifier 1: Save file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MAIL	Mail	***NO**, *YES	Optional
ALWOBJDIF	Allow object differences	Single values: ***NONE**, *ALL Other values (up to 3 repetitions): *AUTL, *OWNER, *PGP	Optional
OMITUSRPRF	User profiles to omit	Single values: ***NONE** Other values (up to 300 repetitions): *Generic name, name*	Optional
SECDTA	Security data	***USRPRF**, *PVTAUT, *PWDGRP, *DCM	Optional
OUTPUT	Output	***NONE**, *OUTFILE	Optional
OPTFILE	Optical file	*Path name*, *****	Optional
SAVASPDEV	Saved from ASP device	*Name*, ***ANY**, *, *SYSBAS, *CURASPGRP	Optional
OUTFILE	File to receive output	*Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	

Device (DEV)

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

This is a required parameter.

Single values

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

Other values

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

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

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

User profile (USRPRF)

Specifies the user profiles to be restored. The user profiles must exist on the media from the Save System (SAVSYS) or Save Security Data (SAVSECDTA) command in order to be restored.

Single values

*ALL   All the user profiles, authorization lists, and authority holders saved by the Save System (SAVSYS) or Save Security Data (SAVSECDTA) command are restored.

*NEW   All the user profiles, authorization lists, and authority holders saved by the Save System (SAVSYS) or Save Security Data (SAVSECDTA) command which currently do not exist on the system are restored.

*NONE
> No user profiles are restored. This value may be specified only if SECDTA(*DCM) is also specified.

Other values (up to 300 repetitions)

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

name   Specify one or more names of specific user profiles that are restored. Both generic names and specific names can be specified in the same command.

Volume identifier (VOL)

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

Single values

*MOUNTED
> The objects are restored from the volumes placed in the device specified for the **Device (DEV)** parameter. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.
>
> **Note:** This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

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

Sequence number (SEQNBR)

Specifies the sequence number of the tape file used for the restore process.

*SEARCH
>> The volume placed in the device is searched for a file containing the saved user profiles; when a match is found, the user profiles are restored. If a match is not found, you must load another tape and try the command again.

>> If the last operation on the device specified *LEAVE for the **End of media option (ENDOPT)** parameter, indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If *LEAVE was not used for the ENDOPT parameter of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

1-16777215
>> Specify the sequence number of the file to be used to restore user profiles.

End of media option (ENDOPT)

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

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

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

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

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

Save file (SAVF)

Specifies the save file used to restore the data.

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

Qualifier 1: Save file

name   Specify the name of save file to be used.

Qualifier 2: Library

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

*CURLIB
>The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

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

Mail (MAIL)

Specifies whether the OfficeVision distribution objects saved from a release before V2R2M0 are restored.

Note: You can specify *YES on this parameter only if you specify *ALL for the **User profile (USRPRF)** parameter.

*NO   Distribution objects that are part of your mail are not restored along with restoring the user profile.

*YES   Distribution objects that are part of your mail are restored along with restoring the user profile if the save data was created before release V2R2M0. Otherwise, no distribution objects are restored. For saved distribution objects created on V2R2M0 or later, specify DLO(*MAIL) on the Restore Document Library Objects (RSTDLO) command to restore your mail.

Allow object differences (ALWOBJDIF)

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

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

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

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

Single values

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

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

Other values (up to 3 repetitions)

*AUTL

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

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

*OWNER

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

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

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

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

Note: The *PGP value does not apply to user profiles. User profiles with primary group
differences are always restored.

User profiles to omit (OMITUSRPRF)

Specifies user profiles to be omitted from the restore.

Single values

*NONE

None of the user profiles will be omitted from the restore.

Other values (up to 300 repetitions)

generic-name

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

name   Specify one or more names of specific user profiles that are to be omitted from the restore. Both
generic names and specific names can be specified in the same command.

Security data (SECDTA)

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 SAVSECDTA command 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. This value cannot be specified if USRPRF(*NEW) is specified.

*PWDGRP
The passwords and group linkages for the specified user profiles are restored with the user profiles. This value cannot be specified if USRPRF(*ALL) is specified.

*DCM Only the internal objects required by Digital Certificate Manager (DCM) are restored. No user profiles are restored. If this value is specified then USRPRF(*NONE) must also be specified.

Output (OUTPUT)

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

*NONE
No output is created.

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

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

Optical file (OPTFILE)

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

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

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

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

Saved from ASP device (SAVASPDEV)

Specifies the name of the auxiliary storage pool (ASP) device from which private authority information was saved. The private authority information is restored for later use by the Restore Authority (RSTAUT) function.

*ANY   The private authority information saved from all ASPs included in the save operation is restored.

*   The private authority information saved from the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group is restored.

*SYSBAS
> The private authority information saved from the system ASP and all basic user ASPs is restored.

*CURASPGRP
> If the current thread has an ASP group, the private authority information saved from all independent ASPs in the ASP group is restored.

name   Specify the name of the ASP device from which private authority information was saved.

File to receive output (OUTFILE)

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

Qualifier 1: File to receive output

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

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Output member options (OUTMBR)

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

Element 1: Member to receive output

*FIRST

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

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

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

Element 2: Replace or add records

*REPLACE

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

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

Examples

Example 1: Restoring All Profiles

```
RSTUSRPRF   DEV(TAP01)  SEQNBR(*SEARCH)  ENDOPT(*REWIND)
```

This command restores all user profiles contained on the tape currently put on the tape drive named TAP01 to the system. The tape is searched for the file, and the tape is rewound on completion or at the end of restore.

Example 2: Restoring Specific User Profiles

```
RSTUSRPRF   DEV(TAP01)  USRPRF(USRA USRB USRC USER*)
```

This command restores user profiles USRA, USRB, and USRC, along with all the user profiles whose names start with USER. The saved version of all the user profiles must exist on the tape placed on tape drive TAP01.

Example 3: Restoring User Profiles from a Save File

```
RSTUSRPRF   DEV(*SAVF) USRPRF(USRX USRY)  SAVF(QGPL/SAVESEC)
```

This command restores user profiles USRX and USRY to the system from the save file SAVESEC in library QGPL.

Example 4: Reporting Information about User Profiles Restored and Not Restored

```
RSTUSRPRF   DEV(TAP01) USRPRF(*ALL)  OUTPUT(*OUTFILE)
            OUTFILE(PRFS92) OUTMBR(FOURQT *ADD)
```

This command restores all user profiles from the tape device TAP01. A list reporting information about user profiles restored and not restored is directed to the output file PRFS92. The output is received in the member FOURQT as an addition to existing information in the member.

Error messages

*ESCAPE Messages

CPD3774

USRPRF(*ALL) required when MAIL(*YES) specified.

CPF2206

User needs authority to do requested function on object.

CPF222E

&1 special authority is required.

CPF370C

Not authorized to ALWOBJDIF parameter.

CPF3709

Tape devices do not support same densities.

CPF3727

Duplicate device &1 specified on device name list.

CPF3728

Device &1 specified with other devices.

CPF3733

&2 &1 in &3 previously damaged.

CPF3738

Device &1 used for save or restore is damaged.

CPF3743

File cannot be restored, displayed, or listed.

CPF3748

Object information for library &1 damaged.

CPF376B

File &1 not found.

CPF3767

Device &1 not found.

CPF3768

Device &1 not valid for command.

CPF3775

Not all user profiles or authority objects restored.

CPF3780

Specified file for library &1 not found.

CPF3782

File &1 in &2 not a save file.

CPF3785

Not all subsystems ended.

CPF3793

Machine or ASP storage limit reached.

CPF3794

Save or restore operation ended unsuccessfully.

CPF3796

Storage limit exceeded for user profile &4.

CPF380C

Library &1 not restored.

CPF3812

Save file &1 in &2 in use.

CPF908A

Requester &1 not enrolled.

CPF9812

File &1 in library &2 not found.

CPF9814

Device &1 not found.

CPF9833

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

CPFB8ED

Device description &1 not correct for operation.

Top

Return from Subroutine (RTNSUBR)

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

Threadsafe: Yes

The Return from Subroutine (RTNSUBR) command returns control to the command immediately following the Call Subroutine (CALLSUBR) command that called the subroutine. The RTNSUBR command must be used within a subroutine, and multiple RTNSUBR commands may be used in a subroutine. If a value is to be returned, such as an error code, the returned value must be able to be stored into a 4-byte signed integer CL variable. The **Return value (RTNVAL)** parameter can be a variable or constant. If no RTNVAL parameter is defined, the value will default to zero.

Restrictions:
- This command is valid only within a CL subroutine.

Parameters

Keyword	Description	Choices	Notes
RTNVAL	Return value	*Integer*, **0**	Optional

Return value (RTNVAL)

Specifies the value to be returned from the subroutine. This can be a variable or constant that can be stored in a 4-byte signed integer CL variable. If the Call Subroutine (CALLSUBR) command that called the subroutine specified a **CL variable for returned value**, the variable is set to this value.

0   The subroutine returns a zero.

integer-constant
    Specify the constant integer value for the return value.

CL-variable-name
    Specify the name of the CL variable to contain the return value from the called subroutine. This must be a signed integer CL variable.

Examples

```
PGM
 :
SUBR SUBR1
 :
```

```
IF  (&A *LT 30)  THEN(RTNSUBR RTNVAL(-1))
 :
ENDSUBR
ENDPGM
```

This CL subroutine is exited by the RTNSUBR command if &A is less than 30, with a return value of -1.

<div align="right">Top</div>

Error messages

None

<div align="right">Top</div>

Retrieve Auth List Entry (RTVAUTLE)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Authorization List Entry (RTVAUTLE) command is used in a CL program or REXX procedure to retrieve the authorities that a user has on the authorization list. It can be used with the Change Authorization List Entry command to change the user's authorities to include new authorities in addition to the existing authorities for the user.

The authorization list name and user name must be specified. The variables for each of the authorities the user might have are returned blank if the user does not have the authority; they are returned with the correct value for the Change Authorization List Entry (CHGAUTLE) command if the user has the authority. The values are returned in the specified variables for the specified user.

Users with authorization list management (*AUTLMGT) authority, or who own the authorization list, can retrieve authority for any user on the list. Other users can get their own authorities or the authority of *PUBLIC.

Parameters

Keyword	Description	Choices	Notes
AUTL	Authorization list	*Name*	Required, Positional 1
USER	User	*Name*, *PUBLIC	Required, Positional 2
ALL	CL var for *ALL (10)	*Character value*	Optional
CHANGE	CL var for *CHANGE (10)	*Character value*	Optional
USE	CL var for *USE (10)	*Character value*	Optional
EXCLUDE	CL var for *EXCLUDE (10)	*Character value*	Optional
OBJALTER	CL var for *OBJALTER (10)	*Character value*	Optional
OBJEXIST	CL var for *OBJEXIST (10)	*Character value*	Optional
OBJMGT	CL var for *OBJMGT (10)	*Character value*	Optional
OBJOPR	CL var for *OBJOPR (10)	*Character value*	Optional
OBJREF	CL var for *OBJREF (10)	*Character value*	Optional
READ	CL var for *READ (10)	*Character value*	Optional
ADD	CL var for *ADD (10)	*Character value*	Optional
UPDATE	CL var for *UPD (10)	*Character value*	Optional
DELETE	CL var for *DLT (10)	*Character value*	Optional
EXECUTE	CL var for *EXECUTE (10)	*Character value*	Optional
AUTLMGT	CL var for *AUTLMGT (10)	*Character value*	Optional

Authorization list (AUTL)

Specifies the authorization list that the user's authorities come from.

This is a required parameter.

name   Specify the name of the authorization list to be changed.

User (USER)

Specifies the user whose information is to be retrieved. If a variable is specified, it must be a 10-character field which contains a user name or the value *PUBLIC.

This is a required parameter.

*PUBLIC
> The information returned in the specified parameter is for the users who do not have any specific authority to the authorization list, and whose groups do not have any specific authority to the authorization list.

name   Specify the name of the user profile of the user whose information is to be retrieved.

CL var for *ALL (10) (ALL)

Specifies the name of a variable that is used to return the special value of *ALL, if the user has *ALL authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

CL var for *CHANGE (10) (CHANGE)

Specifies the name of a variable that is used to return the special value of *CHANGE, if the user has change authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

CL var for *USE (10) (USE)

Specifies the name of a variable that is used to return the special value of *USE, if the user has use authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

CL var for *EXCLUDE (10) (EXCLUDE)

Specifies the name of a variable that is used to return the special value of *EXCLUDE, if the user has that authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

CL var for *OBJALTER (10) (OBJALTER)

Specifies the name of a variable that is used to return *OBJALTER if the user has *OBJALTER authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *OBJALTER authority.

Blanks are returned for the variable if the user does not have this authority.

CL var for *OBJEXIST (10) (OBJEXIST)

Specifies the name of a variable that is used to return the special value of *OBJEXIST, if the user has that authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

CL var for *OBJMGT (10) (OBJMGT)

Specifies the name of a variable that is used to return the special value of *OBJMGT, if the user has object management authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

CL var for *OBJOPR (10) (OBJOPR)

Specifies the name of a variable that is used to return the special value of *OBJOPR, if the user has object operation authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

CL var for *OBJREF (10) (OBJREF)

Specifies the name of a variable that is used to return *OBJREF if the user has *OBJREF authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *OBJREF authority.

Blanks are returned for the variable if the user does not have this authority.

CL var for *READ (10) (READ)

Specifies the name of a variable that is used to return the special value of *READ, if the user has read authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

CL var for *ADD (10) (ADD)

Specifies the name of a variable that is used to return the special value of *ADD, if the user has add authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

CL var for *UPD (10) (UPDATE)

Specifies the name of a variable that is used to return the special value of *UPD, if the user has update authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

CL var for *DLT (10) (DELETE)

Specifies the name of a variable that is used to return the special value of *DLT, if the user has delete authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

CL var for *EXECUTE (10) (EXECUTE)

Specifies the name of a variable that is used to return *EXECUTE if the user has *EXECUTE authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *EXECUTE authority.

Blanks are returned for the variable if the user does not have this authority.

CL var for *AUTLMGT (10) (AUTLMGT)

Specifies the name of a variable that is used to return the special value of *AUTLMGT, if the user has authorization list management authority.

Blanks are returned for the variable if the user does not have this authority.

Examples

```
ADDAUTLE   AUTL(PAYROLL)  USER(TOM)
           AUT(*OBJOPR *READ *UPD *AUTLMGT)
```

When user Smith calls a CL program containing the following:

```
DCL     &CHG        *CHAR 10
DCL     &ALL        *CHAR 10
DCL     &USE        *CHAR 10
DCL     &EXCL       *CHAR 10
DCL     &OBJOP      *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(&OBJOP)  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

*ESCAPE Messages

CPF22A7

   User &1 not on authorization list &2, no authorities retrieved.

CPF22A8

   Not authorized to retrieve authorities for user &1.

CPF2204

   User profile &1 not found.

CPF2283

   Authorization list &1 does not exist.

CPF2289

   Unable to allocate authorization list &1.

Retrieve Backup Options (RTVBCKUP)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Backup (RTVBCKUP) command is used in a CL program or REXX procedure allows the user to retrieve the options in one of the predefined backups into CL variables. More information on backup is in the Backup and Recovery book, SC41-5304.

Parameters

Keyword	Description	Choices	Notes
BCKUPOPT	Backup options	*DAILY, *WEEKLY, *MONTHLY	Required, Positional 1
DEV	CL var for DEV (43)	*Character value*	Optional
TAPSET	CL var for TAPSET (34)	*Character value*	Optional
CLRTAP	CL var for CLRTAP (4)	*Character value*	Optional
SBMJOB	CL var for SBMJOB (4)	*Character value*	Optional
CHGONLY	CL var for CHGONLY (4)	*Character value*	Optional
PRTRPT	CL var for PRTRPT (4)	*Character value*	Optional
LIB	CL var for LIB (10)	*Character value*	Optional
FLR	CL var for FLR (10)	*Character value*	Optional
DIR	CL var for DIR (10)	*Character value*	Optional
SECDTA	CL var for SECDTA (4)	*Character value*	Optional
CFG	CL var for CFG (4)	*Character value*	Optional
MAIL	CL var for MAIL (4)	*Character value*	Optional
CAL	CL var for CAL (4)	*Character value*	Optional
EXITPGM	CL var for EXITPGM (10)	*Character value*	Optional
EXITPGMLIB	CL var for EXITPGMLIB (10)	*Character value*	Optional

Backup options (BCKUPOPT)

Specifies the backup options to be retrieved.

This is a required parameter.

*DAILY
> The daily backup options are retrieved.

*WEEKLY
> The weekly backup options are retrieved.

*MONTHLY
> The monthly backup options are retrieved.

CL var for DEV (43) (DEV)

Specifies the name of the CL variable that receives the device value. The variable has a minimum length of 43 characters. The value returned is a character string of four 10-character device names, separated by blanks.

CL var for TAPSET (34) (TAPSET)

Specifies the name of the CL variable that receives the tape set names. The variable has a minimum length of 34 characters (seven 4-character tape set names, separated by blanks).

CL var for CLRTAP (4) (CLRTAP)

Specifies the name of the CL variable that receives the indicator for clearing the tape for backup. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

CL var for SBMJOB (4) (SBMJOB)

Specifies the name of the CL variable that receives the indicator of whether the backup is run as a batch job. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

CL var for CHGONLY (4) (CHGONLY)

Specifies the name of the CL variable that receives the indicator for saving changed objects only. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

CL var for PRTRPT (4) (PRTRPT)

Specifies the name of the CL variable that receives the indicator for printing a report of saved objects. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

CL var for LIB (10) (LIB)

Specifies the name of the CL variable that receives the value specifying the libraries to save with this backup. The variable must have a minimum length of 10 characters. A value of *ALLUSR, *FROMLIST, or *NONE is returned.

CL var for FLR (10) (FLR)

Specifies the name of the CL variable that receives the value specifying the folders to save with this backup. The variable must have a minimum length of 10 characters. A value of *ALL, *FROMLIST, or *NONE is returned.

CL var for DIR (10) (DIR)

Specifies the name of the CL variable that receives the value specifying the user directories to save with this backup. The variable must have a minimum length of 10 characters. A value of *ALLUSR or *NONE is returned.

CL var for SECDTA (4) (SECDTA)

Specifies the name of the CL variable that receives the indicator for saving security data. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

CL var for CFG (4) (CFG)

Specifies the name of the CL variable that receives the indicator for saving configuration data. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

CL var for MAIL (4) (MAIL)

Specifies the name of the CL variable that receives the indicator for saving OfficeVision mail. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

CL var for CAL (4) (CAL)

Specifies the name of the CL variable that receives the indicator for saving OfficeVision calendars. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

CL var for EXITPGM (10) (EXITPGM)

Specifies the name of the CL variable that receives the name of the user program to call before and after the backup is run. The variable must have a minimum length of 10 characters. If no exit program is specified, *NONE is returned.

CL var for EXITPGMLIB (10) (EXITPGMLIB)

Specifies the name of the CL variable that receives the name of the library that contains the exit program. The variable must have a minimum length of 10 characters. If no exit program is specified, blanks are returned. If *LIBL is returned, the program uses the library list.

Examples

```
RTVBCKUP   BCKUPOPT(*DAILY)   SBMJOB(&SBMJOBVAR)   LIB(&LIBVAR)
```

This command retrieves the SBMJOB and LIB values for the daily backup into the CL variables SBMJOBVAR and LIBVAR respectively.

Error messages

*ESCAPE Messages

CPF1EE3
> Not authorized to backup options.

CPF1E6C
> Backup options in use.

CPF1E67
> Backup options and library backup list damaged.

CPF1E99
> Unexpected error occurred.

Retrieve Binder Source (RTVBNDSRC)

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

The Retrieve Binder Source (RTVBNDSRC) command can be used to retrieve the exports from a set of modules, a service program, or both, and place them (along with the binder language statements needed for the exports) in a specified file member. This file member can later be used as input to the Create Service Program (CRTSRVPGM) command SRCMBR parameter. After the binder language has been retrieved into a source file member, you can edit the binder language to make changes as needed.

By default, the CRTSRVPGM command has a binder language file specified on the EXPORT and SRCFILE parameters to identify the exports from the service program. The RTVBNDSRC command can be useful in helping you automatically create this binder language.

Restrictions:

- You must have use (*USE) authority to the Create Source Physical File (CRTSRCPF) command, if the file does not exist.
- You must have *USE authority to the Reorganize Physical File Member (RGZPFM) command.
- You must have *USE authority to the Add Physical File Member (ADDPFM) command, if the member does not exist.
- You must have *USE authority to the modules or service program from which the exports are being retrieved.
- You must have execute (*EXECUTE) authority to the libraries in which the modules or service program exist.
- If the source file and member to receive the binder language exist, you must have change (*CHANGE) authority and either object alter (*OBJALTER) or object management (*OBJMGT) authority to the file, and *EXECUTE authority to the library that contains the file.
- If the source file exists but the source member needs to be created, you must have *CHANGE authority and either *OBJALTER or *OBJMGT authority to the file, and *EXECUTE, read (*READ) and add (*ADD) authorities to the library that contains the file.
- If the source file and member need to be created, you must have *EXECUTE, *READ and *ADD authorities to the library.

Parameters

Keyword	Description	Choices	Notes
MODULE	Module	Values (up to 300 repetitions): *Qualified object name*	Optional, Positional 1
	Qualifier 1: Module	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB, *USRLIBL	
SRVPGM	Service program	*Qualified object name*	Optional
	Qualifier 1: Service program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Keyword	Description	Choices	Notes
SRCFILE	Export source file	*Qualified object name*	Optional
	Qualifier 1: Export source file	*Name*, **QSRVSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Export source member	*Name*, ***DFT**	Optional
MBROPT	Replace or add records	*ADD, ***REPLACE**	Optional

Module (MODULE)

Specifies the list of modules from which to retrieve the exported symbols. If duplicate module and library specifications are found, only the first instance of the duplicate module and library is used.

At least one value must be specified for this parameter or the **Service program (SRVPGM)** parameter.

You can specify up to 300 values for this parameter.

Qualifier 1: Module

*ALL The exported symbols from all of the modules in the specified library are retrieved.

generic-name
>   Specify a generic module name from which to retrieve the exported symbols. All modules that have names with the same prefix in the specified library or libraries are used. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name.

name Specify the name of the module from which to retrieve the exported symbols.

Qualifier 2: Library

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

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

*USRLIBL
>   Only the libraries in the user portion of the job's library list are searched.

name Specify the name of the library to be searched.

Service program (SRVPGM)

Specifies the service program from which to retrieve the exported symbols. At least one value must be specified for this parameter or the **Module (MODULE)** parameter.

Qualifier 1: Service program

name Specify the name of the service program from which to retrieve the exported symbols.

Qualifier 2: Library

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

*CURLIB

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

name   Specify the name of the library to be searched.

Export source file (SRCFILE)

Specifies the source file that is to hold the binder language for the exported symbols. If the source file does not exist, it is created.

Notes:

- Only a database physical file of type *SRC can be specified. Distributed data management (DDM) files are not supported.
- If the source file to receive the binder language exists, its record length must be a minimum of 92 bytes.

Qualifier 1: Export source file

QSRVSRC

> The source file name is QSRVSRC.

name   Specify the name of the source file.

Qualifier 2: Library

*LIBL   All libraries in the job's library list are searched until the first match is found. If a source file by the name specified is not found in the library list, it is created in the current library. If there is no current library, the QGPL library is used.

*CURLIB

> The current library for the job is searched. If a source file by the name specified does not exist, it is created in the current library. If there is no current library, the QGPL library is used.

name   Specify the name of the library to be searched. If a source file by the name specified is not found in this library, the source file is created in this library.

Export source member (SRCMBR)

Specifies the member in the source file that is to hold the binder language for the exported symbols. Only one source member will contain the binder language.

If the member does not exist in the source file specified, the member is created.

*DFT   The name of the source file member is taken from the value specified for the **Service program (SRVPGM)** parameter, if a service program is specified. Otherwise, the name of the source file member is taken from the value specified for the **:Module (MODULE)** parameter,

- If only one module is specified, the name of that module is the member name used.
- If more than one module is specified, the name of the first module specified is used.

- If the value *ALL or a generic name is specified, the module name of the first occurrence found is the source member name used.

name   Specify the name of the member that will contain the generated source binder language.

Replace or add records (MBROPT)

Specifies whether the generated binder language statements are replaced or added to the existing statements.

*REPLACE
> The system clears the existing member and adds the new records.

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

> **Note:** If the member already exists and, for example, already contains STRPGMEXP and ENDPGMEXP statements, the member may contain multiple STRPGMEXP and ENDPGMEXP statements in the binder language at the end of this operation. You must edit these multiple statements in order to use the binder language with the Create Service Program (CRTSRVPGM) command.

Examples

```
RTVBNDSRC   MODULE(MYLIB/*ALL)
            SRCFILE(MYLIB/MYBINDFILE)  MBROPT(*ADD)
```

This command retrieves the exports from all modules in the library MYLIB, and places them in the source member with the name of the first module found. If this source member does not exist in the file MYBINDFILE in the library MYLIB, it is created. The export statements are added to the end of the member. If multiple start and end program export statements exist in the file when this command is ended, the source member must be edited before it is used to create a service program. Either the extra STRPGMEXP, ENDPGMEXP statements can be removed, or the PGMLVL parameter can be added to the STRPGMEXP statements, if some of the export blocks are for previous versions of the service program.

Error messages

*ESCAPE Messages

CPF5CA5
> Record length too small for data base source file.

CPF5D06
> Not authorized to library &2, or file &1 in library &2, or member &3.

CPF5D08
> File &1 in library &2 not supported file type.

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

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

CPF9803

Cannot allocate object &2 in library &3.

CPF9810

Library &1 not found.

CPF9820

Not authorized to use library &1.

CPF9832

Function not supported for DDM file &2.

CPF9834

Input file &1 in library &2 is not a source file.

CPF9848

Cannot open file &1 in library &2 member &3.

CPF9899

Error occurred during processing of command.

Top

Retrieve Configuration Source (RTVCFGSRC)

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

The Retrieve Configuration Source (RTVCFGSRC) command is used to retrieve the CL source statements which describe existing configuration objects. These source statements are placed into a source file member. The CL source may be used to re-create the configuration objects.

Parameters

Keyword	Description	Choices	Notes
CFGD	Configuration description	Single values: *ALL Other values (up to 256 repetitions): *Generic name, name*	Required, Positional 1
CFGTYPE	Type	*ALL, *NWSD, *NWID, *LIND, *CTLD, *DEVD, *MODD, *COSD, *CNNL, *NTBD	Required, Positional 2
SRCFILE	Source file	*Qualified object name*	Optional, Positional 3
	Qualifier 1: Source file	Name, **QCLSRC**	
	Qualifier 2: Library	Name, ***LIBL**, *CURLIB	
SRCMBR	Source member	Name, ***CFGD**	Optional, Positional 4
RTVOPT	Retrieve option	***NET**, *OBJ	Optional
MBROPT	Member option	*ADD, ***REPLACE**	Optional
TEXT	Text 'description'	*Character value*, ***CFGDTXT**, *BLANK	Optional

Configuration description (CFGD)

Specifies the names of configuration objects to be retrieved.

This is a required parameter.

*ALL   All configuration objects of the specified configuration type (CFGTYPE parameter) are retrieved.

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

configuration-description-name
>   Specify the user-defined name of the configuration description.

Up to 256 names can be specified. These can be any combination of configuration object and generic names.

You can enter multiple values for this parameter.

Type (CFGTYPE)

Specifies the type of configuration object to be retrieved.

*ALL   All network servers, network interfaces, lines, controllers, devices, connection lists, modes, classes-of-service, and NetBIOS descriptions matching the specified names are retrieved in the following order:

    1. Connection Lists

    2. Network server descriptions

    3. Network Interfaces

    4. Non-TDLC line descriptions

    5. Non-TDLC controller descriptions

    6. TDLC line descriptions

    7. TDLC controller descriptions

    8. Device descriptions

    9. Mode descriptions

    10. Class-of-service descriptions

    11. NetBIOS descriptions

    12. SWTCTLLST for line descriptions

    13. SWTLINLST for controller descriptions

    14. SWTNWILST for line descriptions

    15. Printer for remote displays

*NWSD

All network server descriptions matching the specified names are retrieved.

*NWID

All network interface descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

*LIND

All line descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

*CTLD

All controller descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

*DEVD

All device descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

*MODD

All mode descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

*COSD

All class-of-service descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

*CNNL
>All connection lists that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

*NTBD
>All NetBIOS descriptions matching the specified names are retrieved.

Source file (SRCFILE)

Specifies the name and library of the previously created database source file that contains the source file member.

The possible **source-file** values are:

QCLSRC
>The source file named QCLSRC is used.

source-file
>Specify the name of a source file.

The possible library values are:

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

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

library-name
>Specify the library where the source file is located.

Source member (SRCMBR)

Specifies the name of the database source file member into which the CL source statements are written.

*CFGD
>The name of the source member is the configuration object name if there is only one name specified and the name is not generic or *ALL. Otherwise, the name is CFGSRC.

source-member-name
>Specify the name of the source file member that contains the CL source statements.

Retrieve option (RTVOPT)

Specifies which attachment information is retrieved for the specified objects.

*NET For network interfaces, each network interface description and the descriptions of its attached nonswitched lines, controllers, and devices are retrieved. For lines, each line description and the descriptions of its attached nonswitched controllers and devices are retrieved. For controllers, each controller description and the descriptions of its attached devices are retrieved. For network interfaces, lines, and controllers, switched attachment information is also retrieved.

*OBJ Each specified configuration object description is retrieved with all attachment information.

RTVOPT is ignored if the CFGTYPE parameter is *ALL.

Member option (MBROPT)

Specifies whether the new records replace or are added to the existing records.

*REPLACE

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

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

Text 'description' (TEXT)

Specifies the text that briefly describes the object.

*CFGDTXT

The text description is the same as the text description of the object specified by the CFGD parameter unless there is more than one object specified or the name is a generic name. In this case, *BLANK is used for the text description.

*BLANK

No text is specified.

'description'

Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

```
RTVCFGSRC   CFGD(CTL*)   CFGTYPE(*CTLD)
            SRCMBR(CTLS)   RTVOPT(*OBJ)
```

This command places CL source statements in the file member CTLS in the source file QCLSRC. These source statements can be used to re-create object descriptions for all existing controllers with names beginning with CTL.

Error messages

*ESCAPE Messages

CPF2207

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

CPF263E

File member &1 contains its maximum sequence number.

CPF263F

No objects found.

CPF264A

Record length of &1 in library &2 is too small.

CPF264C

Source file member &1 not found.

CPF9810

Library &1 not found.

CPF9820

Not authorized to use library &1.

CPF9847

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

CPF9848

Cannot open file &1 in library &2 member &3.

CPF9849

Error while processing file &1 in library &2 member &3.

Top

Retrieve Configuration Status (RTVCFGSTS)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Configuration Status (RTVCFGSTS) command provides a CL program with a numeric value that represents the status of a configuration object.

Parameters

Keyword	Description	Choices	Notes
CFGD	Configuration description	*Name*	Required, Positional 1
CFGTYPE	Type	*NWI, *NWS, *LIN, *CTL, *DEV	Required, Positional 2
STSCDE	CL variable for status code	*Decimal number*	Required, Positional 3

Configuration description (CFGD)

Specifies the name of the configuration description for which the status will be retrieved.

This is a required parameter.

Type (CFGTYPE)

Specifies the type of description for which you want to retrieve status.

*NWS   The object is a network server description.

*NWI   Status for a network interface is retrieved.

*LIN   Status for a line is retrieved.

*CTL   Status for a controller is retrieved.

*DEV   Status for a device is retrieved.

This is a required parameter.

43

CL variable for status code (STSCDE)

Specifies the name of a variable which will contain the retrieved status. In a CL program, this should be a decimal variable of length (5 0).

The possible values which can be returned are:

Value   Definition

0   VARIED OFF - The system is not using the description.

10   VARY OFF PENDING - The description is being varied off. During this time the system may be taking down the tasks which managed the resource, and so on.

20   VARY ON PENDING - The description is being varied on. During this time the system may be putting tasks in place to manage the resource, downloading a program to an I/O processor, communicating with data circuit-terminating equipment (DCE), and so on.

30   VARIED ON - The tasks that manage the network interface, network server, line, controller, or device have been put in place by the system, and the system has the capability to communicate with it.

32   VARY ON/CNN PENDING - The first of a pair of OptiConnect controllers is varied on but its attached device is not yet in a varied on state.

40   CONNECT PENDING - This status is only valid for switched SDLC, IDLC, BSC, or asynchronous lines. The line is in this status while waiting for the switched connection to be established; this can be either a dial or an answer connection.

50   SIGNON DISPLAY - This status is only valid for display devices. The system is preparing the device to receive the sign-on display, or sending the sign-on display, or the actual sign-on display is at the display station.

51   ACTIVE/CNN PENDING - The first of a pair of OptiConnect controllers and its attached device are varied on and waiting for the OptiConnect path to be established.

60   ACTIVE - The object is successfully placed in VARIED ON status. In addition: for network interfaces and the network servers, one or more attached lines is in a VARY ON PENDING status or higher; for lines, one or more attached controllers is in a VARY ON PENDING status or higher; for controllers, one or more attached devices is in a VARY ON PENDING status or higher; for devices, active status varies depending on the type of device — more information is in the Communications Configuration book, SC41-5401 book. A display device which is at a second sign-on display as a result of pressing the system request key will be considered ACTIVE.

63   ACTIVE READER - The device is in use by a spool reader.

66   ACTIVE WRITER - The device is in use by a spool writer.

67   AVAILABLE - The independent auxiliary storage pool (ASP) device is available for use without function restrictions.

70   HELD - This status is only valid for Device Descriptions. The user or the system held the communications device to prevent it from communicating. The Release Communications Device (RLSCMNDEV) command can be used to release the device.

80   RCYPND - Error recovery is pending for the line, controller, or device. A message indicating what error occurred appears on the QSYSOPR message queue.

90   RCYCNL - Error recovery is canceled for the network interface, line, controller, or device. An error occurred, and the operator replied with a C (to cancel error recovery) to a message, or the operator used a command (ENDNWIRCY, ENDLINRCY, ENDCTLRCY, ENDDEVRCY) to end error recovery.

95 SYSTEM REQUEST - The display device has been requested by the system and its associated job has been suspended. This occurs as a result of a user pressing the System Request key.

100 FAILED - An error occurred for the network interface, network server, line, controller, or device that can be recovered only by varying off and on again.

103 FAILED READER - An error occurred for the device while in use by a spool reader.

106 FAILED WRITER - An error occurred for the device while in use by a spool writer.

107 SHUTDOWN - The NWSD was shut down using an Application Program Interface (API).

110 DIAGNOSTIC MODE - The network interface, network server, 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). Further information may be in the vertical licensed internal code (VLIC) logs. The description must be deleted and created once more before it can be used again.

112 LOCKED - The actual status of the resource cannot be determined because another job has an exclusive lock on the description. Retry at a later time, or use the Work With Object Lock (WRKOBJLCK) command to determine which job has the lock on the description.

113 UNKNOWN - The status indicator of the description cannot be determined. This is a system error condition. Use the Dump Object (DMPOBJ) command to dump the contents or attributes of the description to a spooled printer file, and contact your IBM representative.

This is a required parameter.

More information is in the Communications Configuration book, SC41-5401.

<div align="right">Top</div>

Examples

```
RTVCFGSTS   CFGD(ND01)   CFGTYPE(*LIN)   STSCDE(&STSCODE)
```

This command retrieves the configuration status of the line configuration description ND01 for use in the CL variable &STSCODE.

<div align="right">Top</div>

Error messages

*ESCAPE Messages

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

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

<div align="right">Top</div>

Retrieve C Locale Description (RTVCLDSRC)

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

The Retrieve C/400* Locale Description Source (RTVCLDSRC) command retrieves the locale that you specified on the SRCFILE parameter. The *SYSVAL locale can be retrieved and used as a model to create a tailored locale description. The locale commands and formats that you use in your locale source are outlined in the C/400 User's Guide. Once your locale source is complete, use the CRTCLD command to process the description.

Error messages for RTVCLDSRC

*ESCAPE Messages

PSE1708
   Member &3 is not found in the specified file and library.

PSE1709
   File &1 is not found in library &2.

PSE1739
   The necessary storage could not be allocated.

PSE1751
   Source &1 in file &3 was not retrieved.

Parameters

Keyword	Description	Choices	Notes
CLD	Locale name	*Qualified object name*	Required, Positional 1
	Qualifier 1: Locale name	*Name*, *SYSVAL	
	Qualifier 2: Library	*Name*, ***CURLIB**	
SRCFILE	Source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Source file	*Name*, **QCLDSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***CLD**	Optional, Positional 3
TEXT	Text description	*Character value*, ***SRCMBRTXT**, *BLANK	Optional, Positional 4

Locale name (CLD)

Specifies the C/400 locale description that is to be retrieved. You can retrieve the system values if you use *SYSVAL, instead of specifying a locale name and library.

locale name
> Enter the name of the locale description.

*SYSVAL
> The system default values are used as the basis for the retrieved locale description source. The default locale description values are used for any locale description information that does not have a corresponding system value.

The possible library values are:

*CURLIB
> The current library will be searched for the source file specified. If you have not specified a current library, QGPL will be used.

library-name
> Enter the name of the library containing the locale description.

<div align="right">Top</div>

Source file (SRCFILE)

Specifies the name and library of the file that receives the retrieved C/400 locale description source. If the specified library does not exist, the process ends. If the specified file does not exist, it is created.

QCLDSRC
> The default source file name for the C locale description source. Use this default if you want to browse your source and its associated listing using SEU.

source-file-name
> Enter the name of the file where you want to store the C locale description source.

The possible library values are:

*LIBL The system searches the library list for the library that contains the locale description source file.

*CURLIB
> The current library is used to store the retrieved locale. If you have not specified the current library, QGPL will be used.

library-name
> Enter the name of the library to store the retrieved locale.

<div align="right">Top</div>

Source member (SRCMBR)

Specifies the source file member that will store the C/400 locale description source after processing. If the specified member does not exist, it is created.

*CLD Specifies that the C locale description name specified by the CLD parameter is used as the source file member name. For example, if you retrieved the system values, the locale source is stored in a member called SYSVAL.

source-file-member-name
> Enter a name for the source file member that will store the retrieved C locale description.

<div align="right">Top</div>

Text description (TEXT)

Specifies the text you want associated with the source file member you are processing.

*SRCMBRTXT
> Specifies that the same text description you supplied for the C locale description is used.

*BLANK
> No text appears.

'description'
> Enter the descriptive text to accompany the retrieved C locale description. You can enter up to 50 characters of text.

<div align="right">Top</div>

Examples

None

<div align="right">Top</div>

Error messages

*ESCAPE Messages

PSE1708
> Member &3 is not found in the specified file and library.

PSE1709
> File &1 is not found in library &2.

PSE1739
> The necessary storage could not be allocated.

PSE1751
> Source &1 in file &3 was not retrieved.

<div align="right">Top</div>

Retrieve Cleanup (RTVCLNUP)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Cleanup (RTVCLNUP) command retrieves a cleanup operation value for use in a CL or REXX program. The value is returned (copied) to the specified CL variable in the program.

Parameters

Keyword	Description	Choices	Notes
ALWCLNUP	CL var for ALWCLNUP (4)	*Character value*	Optional
STRTIME	CL var for STRTIME (10)	*Character value*	Optional
USRMSG	CL var for USRMSG (5)	*Character value*	Optional
SYSMSG	CL var for SYSMSG (5)	*Character value*	Optional
CRITSYSMSG	CL var for CRITSYSMSG (5)	*Character value*	Optional
SYSPRT	CL var for SYSPRT (5)	*Character value*	Optional
SYSLOG	CL var for SYSLOG (5)	*Character value*	Optional
JOBQ	CL var for JOBQ name (10)	*Character value*	Optional
JOBQLIB	CL var for JOBQ library (10)	*Character value*	Optional
RUNPTY	CL var for RUNPTY (2 0)	*Decimal number*	Optional
JRNRCVSIZ	CL var for JRNRCVSIZ (7 0)	*Decimal number*	Optional
CALITM	CL var for CALITM (5)	*Character value*	Optional

CL var for ALWCLNUP (4) (ALWCLNUP)

Specifies the name of the CL variable that receives the allow cleanup value. The variable named has a minimum length of 4 characters. A '*YES' is returned if the cleanup operation is allowed to run. Otherwise, an '*NO' is returned.

CL var for STRTIME (10) (STRTIME)

Specifies the name of the CL variable that receives the time cleanup starts each day. The variable named has a minimum length of 10 characters. A special value of '*NONE' or '*SCDPWROFF,' or the start time is returned.

CL var for USRMSG (5) (USRMSG)

Specifies the name of the CL variable that receives the value for cleaning up user messages on user profile message queues. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days user messages are kept before they are deleted is returned.

CL var for SYSMSG (5) (SYSMSG)

Specifies the name of the CL variable that receives the value for cleaning up messages on the QSYSOPR message queue and on work station message queues. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days system messages are kept before they are deleted is returned.

CL var for CRITSYSMSG (5) (CRITSYSMSG)

Specifies the name of the CL variable that receives the value for cleaning up messages on the QSYSMSG message queue. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days critical system messages are kept before they are deleted is returned.

CL var for SYSPRT (5) (SYSPRT)

Specifies the name of the CL variable that receives the value for cleaning up job logs and other system output. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days job logs are kept before they are deleted is returned.

CL var for SYSLOG (5) (SYSLOG)

Specifies the name of the CL variable that receives the value for cleaning up system journals, history files, problem log files, alert database, and program temporary fixes. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days system journals and system logs are kept before they are deleted is returned.

CL var for JOBQ name (10) (JOBQ)

Specifies the name of the CL variable that receives the name of the job queue to which the cleanup batch jobs are submitted. The variable named has a minimum length of 10 characters. The name of the job queue under which cleanup batch jobs are run is returned.

CL var for JOBQ library (10) (JOBQLIB)

Specifies the name of the CL variable that receives the library name of the job queue to which the cleanup batch jobs are submitted. The variable named has a minimum length of 10 characters.

CL var for RUNPTY (2 0) (RUNPTY)

Specifies the name of the CL variable that receives the 2-digit value, ranging from 1 through 99, that is the run (or processing) priority for routing steps that are part of the job. For additional information on run priority, refer to this parameter description in the CHGJOB (Change Job) command. The variable must be a 2-digit decimal variable specified with no decimal positions.

CL var for JRNRCVSIZ (7 0) (JRNRCVSIZ)

Specifies the name of the CL variable that receives the value for the journal receiver size ranging from 1 through 1,919,999 (KB). When the size of the space for the journal receiver is larger than the size specified by this value, Operational Assistant (OA) automatic cleanup function will detach it. The variable must be an 7-digit decimal variable specified with no decimal positions.

CL var for CALITM (5) (CALITM)

In V5R1 and later releases, this parameter is not supported.

Examples

Example 1: Retrieving Number of Days Messages are Kept

```
DCL   VAR(&UMSGDAYS)  TYPE(*CHAR)  LEN(5)
RTVCLNUP   USRMSG(&UMSGDAYS)
```

These commands retrieve the number of days that user messages are kept before being deleted.

Example 2: Retrieving Time Cleanup Operation Starts

```
DCL   VAR(&CLNUPTIME)  TYPE(CHAR)  LEN(10)
RTVCLNUP   STRTIME(&CLNUPTIME)
```

These commands retrieve the time that the cleanup operation starts.

Example 3: Retrieving Run Priority

```
DCL   VAR(&RPTY)  TYPE(*DEC)  LEN(2 0)
RTVCLNUP   RUNPTY(&RPTY)
```

This command retrieves the run priority for the cleanup job. The run priority number is copied into the CL variable &RPTY. The variable must be a 2-digit decimal variable with no decimal positions.

Example 4: Retrieving Journal Receiver Size

```
DCL    VAR(&JRNSIZ) TYPE(*DEC)  LEN(7 0)
RTVCLNUP  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.

<div align="right">Top</div>

Error messages

*ESCAPE Messages

CPF1E2B

Power scheduler and cleanup options not found.

CPF1E33

Cleanup options or power schedule in use by another user.

CPF1E99

Unexpected error occurred.

<div align="right">Top</div>

Retrieve CL Source (RTVCLSRC)

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

The Retrieve CL Source (RTVCLSRC) command is used to retrieve the source statements from an OPM CL program that was 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.

Parameters

Keyword	Description	Choices	Notes
PGM	Program	*Qualified object name*	Required, Positional 1
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCFILE	Source file	*Qualified object name*	Required, Positional 2
	Qualifier 1: Source file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***PGM**	Optional, Positional 3

Program (PGM)

Specifies the OPM CL program whose source is to be retrieved.

Note: RTVCLSRC cannot be used to retrieve the source for ILE CL programs and modules.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the program.

Qualifier 2: Library

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

*CURLIB
> The current library for the thread is used to locate the program. If no library is specified as the current library for the thread, the QGPL library is used.

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

Source file (SRCFILE)

Specifies the database source file into which the CL source statements are to be written. This file must exist when the command is run.

This is a required parameter.

Qualifier 1: Source file

name   Specify the name of the source file.

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the database source file member into which the CL source statements are to be written. If the name is not specified, the CL program name is used. If the member existed before the command was run, it is cleared before any source statements are written into it. If the member did not exist, it is created.

*PGM   The name of the CL program is used as the member name.

name   Specify the name of the source file member that will contain the CL source statements.

Top

Examples

```
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.

Top

Error messages

*ESCAPE Messages

CPF0560
    Program &1 in &2 not a CL program.

CPF0561
    Unable to retrieve CL source from CL program &2.

CPF0562
    File &1 in &2 not a data base source file.

CPF0563
    Record length too small for data base source file.

CPF0564

        Unable to add data base member &3 to file.

CPF0565

        Source from CL program &4 not retrieved.

CPF0566

        Source not available for CL program &1 in &2.

CPF9801

        Object &2 in library &3 not found.

CPF9803

        Cannot allocate object &2 in library &3.

CPF9805

        Object &2 in library &3 destroyed.

CPF9806

        Cannot perform function for object &2 in library &3.

CPF9807

        One or more libraries in library list deleted.

CPF9808

        Cannot allocate one or more libraries on library list.

CPF9809

        Library &1 cannot be accessed.

CPF9810

        Library &1 not found.

CPF9811

        Program &1 in library &2 not found.

CPF9820

        Not authorized to use library &1.

CPF9821

        Not authorized to program &1 in library &2.

CPF9822

        Not authorized to file &1 in library &2.

CPF9830

        Cannot assign library &1.

CPF9848

        Cannot open file &1 in library &2 member &3.

CPF9849

        Error while processing file &1 in library &2 member &3.

Top

Retrieve Current Directory (RTVCURDIR)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Current Directory (RTVCURDIR) command is used in a control language (CL) program to retrieve the name of the current directory into the specified CL variable. An absolute path name containing no symbolic links is retrieved. The length of the name of the current directory is also retrieved.

The CL prompt for this command lists the minimum length for retrieved variables next to the appropriate parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

1. Execute (*X) authority is required to the current directory and the user must have read, execute (*RX) authority to each directory in the path.
2. This command is valid only within a CL program.
3. 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.

Parameters

Keyword	Description	Choices	Notes
RTNDIR	CL var for RTNDIR (9999)	*Character value*	Required, Positional 1
DIRNAMLEN	CL var for DIRNAMLEN (7 0)	*Decimal number*	Required, Positional 2

CL var for RTNDIR (9999) (RTNDIR)

Specifies the name of the CL variable that receives the name of the current directory. The variable must be a character variable. If the current directory name has fewer characters than the variable allows, the value is not padded.

CL var for DIRNAMLEN (7 0) (DIRNAMLEN)

Specifies the name of the CL variable that receives the length (in bytes) of the current directory name. This length can be longer than the length of the character variable to receive the directory name. The variable must be a 7-digit decimal variable specified with no decimal positions.

59

Examples

Example 1: Retrieving the Current Directory

```
RTVCURDIR   RTNDIR(&CD)  DIRNAMLEN(&CDLEN)
```

This command retrieves the name of the current directory and the length of the name of the current directory into the CD and CDLEN variables.

Error messages

*ESCAPE Messages

CPFA085
> Home directory not found for user &1.

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

CPFA0A1
> An input or output error occurred.

CPFA0A9
> Object not found. Object is &1.

Retrieve Directory Information (RTVDIRINF)

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

The Retrieve Directory Information (RTVDIRINF) command is used to collect attributes for directories and files in the Integrated File System. The collected information is stored in database files that are named using the information file prefix specified by the INFFILEPFX parameter. The files are created in the library specified by the INFLIB parameter.

You can run the Print Directory Information (PRTDIRINF) command to print reports using the retrieved directory information.

To get the most accurate results, this command should be run at a time when there is very little activity for files in the specified directory. If SUBTREE(*ALL) is specified, try to run this command when there is very little activity for files in all subdirectories of the specified directory.

Note: If there is more than one member in the files, the results of running this command can be unpredictable.

Restrictions:

- Directory information can be retrieved only for mounted file systems.
- Directory information can be retrieved only for local file systems.
- You must have all object (*ALLOBJ) special authority to run this command.
- Job CCSID value 65535 is not allowed.

Parameters

Keyword	Description	Choices	Notes
DIR	Directory	*Path name*	Required, Positional 1
SUBTREE	Directory subtree	***ALL**, *NONE	Optional
INFFILEPFX	Information file prefix	*Simple name*, ***GEN**	Optional
INFLIB	Information library	*Name*, **QUSRSYS**	Optional

Directory (DIR)

Specifies the path name of the directory for which directory information will be collected.

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

This is a required parameter.

directory-path-name
>Specify the path name of a directory.

Directory subtree (SUBTREE)

Specifies whether or not to collect directory information for subdirectories of the directory specified by the DIR parameter.

*ALL   Directory information for subdirectories of the specified directory will be collected.

*NONE
>Directory information will only be collected for the files in the specified directory. No directory information will be collected for subdirectories.

Information file prefix (INFFILEPFX)

Specifies the file name prefix of the database files where the retrieved directory information is to be stored.

*GEN   The database files will be created with a unique prefix generated by this command. The prefix will begin with QAEZD followed by four digits. The files created to store the collected information will be named using this prefix followed by either the letter 'D' (for the file which contains directory information) or the letter 'O' (for the file that contains information about objects in directories). For example, the first time the command is run with *GEN specified, files QAEZD0001D and QAEZD0001O will be created in the library specified by the INFLIB parameter. Informational message CPI1E30 will be sent to the job log and will contain the names of the files created.

information-file-prefix
>Specify the file prefix to use for creating the database files used to store the collected directory information. The prefix can be up to nine letters. The files created to store the collected information will be named using this prefix followed by either the letter 'D' (for the file which contains directory information) or the letter 'O' (for the file that contains information about objects in directories). For example, if the prefix specified is MYDIR, database files MYDIRD and MYDIRO will be created in the library specified by the INFLIB parameter.

Information library (INFLIB)

Specifies the library where the database files used to store the directory information will be created.

QUSRSYS
>The files will be created in library QUSRSYS.

library-name
>Specify the name of the library to create the database files.

Examples

Example 1: Retrieve Information, Including Subdirectories

```
RTVDIRINF  DIR('/MYDIR/MYDOCS')  SUBTREE(*ALL)
           INFFILEPFX(*GEN) INFLIB(QUSRSYS)
```

This command retrieves directory information about directory /MYLIB/MYDOCS, including information
for all nested subdirectories, and stores it the database files created in library QUSRSYS. The database
files will be created with unique names that begin with 'QAEZD' followed by four digits. If this is the
first time the RTVDIRINF command is run, the file names will be QAEZD0001O and QAEZD0001D.

Example 2: Retrieve Information for Specified Directory Only
```
RTVDIRINF  DIR('/')  SUBTREE(*NO)  INFFILEPFX(MYROOTDIR)
           INFLIB(MYLIB)
```

This command retrieves directory information about the root directory without inspecting nested
subdirectories and stores it in the database files MYROOTDIRO and MYROOTDIRD in library MYLIB. If
database files with either of those names already exist in library MYLIB, an error message will be sent
and no directory information will be retrieved.

Top

Error messages

*ESCAPE Messages

CPFA08E
> More than one name matches pattern.

CPFA093
> Name matching pattern not found.

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

CPFA0A1
> An input or output error occurred.

CPFA0A3
> Path name resolution causes looping.

CPFA0A6
> Number of links exceeds maximum allowed for the file system.

CPFA0A7
> Path name too long.

CPFA0A9
> Object not found. Object is &1.

CPFA0AA
> Error occurred while attempting to obtain space.

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

CPFA0AD
> Function not supported by file system.

CPFA0B2
> No objects satisfy request.

CPF1ED2
> File &1 is in use and cannot be accessed.

CPF1ED4

Not authorized to collect directory information.

CPF1E99

Unexpected error occurred.

Retrieve DLO Authority (RTVDLOAUT)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Document Library Object Authority (RTVDLOAUT) command retrieves the authority assigned to a folder or filed document.

Restrictions:

- A user must be in the system distribution directory entry to retrieve the various authorities.
- A user must have at least use (*USE) authority to the filed document or folder.
- A user with less than all (*ALL) authority to the filed document or folder will only be able to retrieve that user's authority or the owner.
- A user must have *ALL authority to the object, or all object (*ALLOBJ) special authority, or be the owner of the filed document or folder, to retrieve all the authorities.
- A user must have *ALLOBJ special authority to retrieve the *ROOT folder public authority.

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	*Character value*, *DOCID, *LADNTSP, *SYSOBJNAM, *ROOT	Required, Positional 1
FLR	Folder	*Character value*, ***NONE**	Optional, Positional 2
DOCID	Document identifier	*Character value*, ***NONE**	Optional
LADNTSP	LADN timestamp	0000000000000001-FFFFFFFFFFFFFFFF, ***NONE**	Optional
SYSOBJNAM	System object name	*Name*, ***NONE**	Optional
STRUSRAUTE	Starting user authority entry	1-99999, **1**	Optional
OWNER	CL var for OWNER (10)	*Character value*	Optional
AUTL	CL var for AUTL (10)	*Character value*	Optional
SENSITIV	CL var for SENSITIV (20)	*Character value*	Optional
PUBAUT	CL var for PUBAUT (10)	*Character value*	Optional
CHKOUTUSR	CL var for CHKOUTUSR (32)	*Character value*	Optional
ACC	CL var for ACC (220)	*Character value*	Optional
USRAUT	CL var for USRAUT (1020)	*Character value*	Optional
GRPAUT	CL var for GRPAUT (340)	*Character value*	Optional
PGP	CL var for PGP (10)	*Character value*	Optional

Document library object (DLO)

Specifies the name of the document or folder for which authorities are retrieved.

*ROOT

The root folder contains all first-level folders. If folder *ROOT is specified only public authority will be returned by PUBAUT parameter. No other authorities will be returned.

*LADNTSP

The timestamp from the library-assigned document name (LADN) specified on the LADNTSP parameter is used to identify the document or folder.

*SYSOBJNAM

The system object name specified on the SYSOBJNAM parameter is used to identify the document or folder.

name   Specify the user-assigned name of the document or folder.

Folder (FLR)

Specifies the name of the folder where the object specified on the DLO parameter is located.

*NONE

The name of folder that contains the object is not specified, the object is not contained in a folder, or the object is specified using the LADNTSP, or SYSOBJNAM parameter, or *ROOT was specified in the DLO parameter.

name   Specify the name of the folder that contains the object.

Note: FLR(*NONE) must be specified if the object is a first-level folder.

Document identifier (DOCID)

Specifies the library-assigned name of the document or folder.

*NONE

The object is not identified using its document identifier (DOCID)

document-identifier

Specify the document identifier of the document or folder. The document identifier is 24 hexadecimal characters in length in the format YYYYMMDDHHMNSSHSSNSNSNSN, where:

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

LADN timestamp (LADNTSP)

Specifies the LADN timestamp of the document or folder.

*NONE
> The object is not identified using its LADN timestamp.

timestamp
> Specify the LADN timestamp of the document or folder. The LADN timestamp is 16 hexadecimal characters in length in the format YYYYMMDDHHMNSSHS, where:
> - YYYY = year
> - MM = month
> - DD = day
> - HH = hour
> - MN = minute
> - SS = second
> - HS = hundredths of a second

System object name (SYSOBJNAM)

Specifies the system object name.

*NONE
> The object is not identified using its system object name.

name Specify the 10-character system object name of the document or folder.

Starting user authority entry (STRUSRAUTE)

Allows a user to specify the starting user authority entry number to use when retrieving specific user authorities (USRAUT parameter). STRUSRAUTE will enable retrieving specific user authorities for a DLO that has more than 50 specific user authorities. If no CL variable is provided for the USRAUT parameter, this parameter is ignored.

1 User authority will be returned starting with the first specific user authority.

2-99,999
> User authority will be returned starting with the specified entry number. If the value is greater than the number of specific user authorities for the DLO, an error message will be sent and no specific user authorities will be returned.

CL var for OWNER (10) (OWNER)

Specifies the name of a 10-character CL variable used to retrieve the owner of the selected document or folder.

CL var for AUTL (10) (AUTL)

Specifies the name of a 10-character CL variable used to retrieve the authorization list assigned to the selected document or folder. The value *NONE is returned if no authorization list has been assigned.

CL var for SENSITIV (20) (SENSITIV)

Specifies the name of a 20-character CL variable used to retrieve the sensitivity assigned to the selected document or folder.

*NONE
>The document has no sensitivity restrictions.

*PERSONAL
>The document is intended for the user as an individual.

*PRIVATE
>The document contains information that should be accessed only by the owner.

*CONFIDENTIAL
>The document contains information that should be handled according to company procedures.

CL var for PUBAUT (10) (PUBAUT)

Specifies the name of a 10-character CL variable used to retrieve the public authority assigned to the selected document of folder.

*USE   User can view, print, or copy the document or folder.

*CHANGE
>User can perform all operations listed for *USE and can also edit and mark the document for offline storage.

*ALL   User can perform all operations on the document, except change ownership of the document or folder or give themselves authority to work with the document after authority has been revoked.

*EXCLUDE
>All users who are not otherwise authorized to this document or folder are denied access.

USER DEF
>Authority to this document is user-defined and is not one of the system-defined sets of authorities (*ALL, *CHANGE, *USE, *EXCLUDE).

*AUTL
>Authority specified in the authorization list being used by this document should determine public authority.

CL var for CHKOUTUSR (32) (CHKOUTUSR)

Specifies the name of a 32-character CL variable used to retrieve the user profile who has the document checked out and the user profile on whose behalf the document was checked out. If no user has checked out the documented, 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.

CL var for ACC (220) (ACC)

Specifies the name of a 200-character CL variable used to retrieve the access codes assigned to the specified document or folder.

CL var for USRAUT (1020) (USRAUT)

Specifies the name of a 1020-character CL variable used to retrieve the specific user authority assigned to the document or folder. USRAUT will return a maximum of 50 specific authorities per invocation of the RTVDLOAUT command. If the document or folder has more than 50 specific user authorities associated with it, you can use the STRUSRAUTE parameter on subsequent invocations of RTVDLOAUT to return user authorities starting with the specified entry numbers.

Within the 1020-character CL variable are the total number of authorized users, the number of authority entries returned, the starting entry number and the ending entry number.

For example:

RTVDLOAUT DLO(MYDOC) FLR(MYFLR) USRAUT(&RTNUSRAUT) + STRUSRAUTE(1)

In the example above, MYDOC has 55 authorized users. The total number of specific user authorities will be 55. The number of authority entries returned will be 50. The starting authority entry number will be 1. The ending authority entry number will be 51.

CL var for GRPAUT (340) (GRPAUT)

Specifies the name of a 340-character CL variable used to retrieve the group authority assigned to the specified document or folder. Only the groups associated to the user of this command will be returned.

CL var for PGP (10) (PGP)

Specifies the name of a 10-character CL variable used to retrieve the primary group assigned to the specified document or folder. The value *NONE is returned if no primary group has been assigned.

Examples

Example 1: Retrieve the Owner of a Document Library Object

```
RTVDLOAUT   DLO(MYDOC)   FLR(MYFLR)   OWNER(&OWNER)
```

This command retrieves the owner of document library object MYDOC in folder MYFLR and stores the value in the variable &OWNER.

Example 2: Retrieve Private User Authorities for a Document

```
RTVDLOAUT   DLO(MYDOC)   FLR(MYFLR)
            USRAUT(&RTNUSRAUT)   STRUSRAUTE(1)
```

This command retrieves the user authorities for document MYDOC in folder MYFLR and stores the list in the &RTNUSRAUT variable. The list will contain up to 50 privately authorized users and their authorities.

Error messages

*ESCAPE Messages

CPF8AC0
    &1 command failed.

Retrieve DLO Name (RTVDLONAM)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Document Library Object Name (RTVDLONAM) command is used to retrieve an alternate name for a filed document, folder, or distribution document and place the value into the specified variables. This command is valid only within a CL program or REXX exec.

Restrictions:

- A user must have use (*USE) authority to the filed document or folder to retrieve the various forms of the name.
- A user must have all object (*ALLOBJ) special authority to retrieve the various forms of the name for a distribution document.

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	*Character value*, *DOCID, *LADNTSP, *SYSOBJNAM	Required, Positional 1
FLR	Folder	*Character value*, ***NONE**	Optional, Positional 2
DOCID	Document identifier	*Character value*, ***NONE**	Optional
LADNTSP	LADN timestamp	0000000000000001-FFFFFFFFFFFFFFFF, ***NONE**	Optional
SYSOBJNAM	System object name	*Name*, ***NONE**	Optional
OBJCLS	Object class	***DOC**, *FLR, *DST	Optional
RTNDLO	CL var for RTNDLO (12)	*Character value*	Optional
RTNFLR	CL var for RTNFLR (63)	*Character value*	Optional
RTNDOCID	CL var for RTNDOCID (24)	*Character value*	Optional
RTNLADNTSP	CL var for RTNLADNTSP (16)	*Character value*	Optional
RTNOBJNAM	CL var for RTNOBJNAM (10)	*Character value*	Optional
RTNOBJCLS	CL var for RTNOBJCLS (8)	*Character value*	Optional
RTNASP	CL var for RTNASP (2 0)	*Decimal number*	Optional
RTNOVRFLW	CL var for RTNOVRFLW (1)	*Character value*	Optional

Document library object (DLO)

Specifies the document or folder for which a name is to be retrieved.

This is a required parameter.

*DOCID

The document or folder is identified using its library-assigned document name specified in the DOCID parameter. The **Folder (FLR)** parameter must be *NONE.

*LADNTSP

The document or folder is identified using its library-assigned document name (LADN) timestamp specified in the LADNTSP parameter. The FLR parameter must be *NONE.

*SYSOBJNAM

The document or folder is identified using its system object name specified in the SYSOBJNAM parameter. The FLR parameter must be *NONE.

name   Specify the user-assigned name of the document or folder. The folder containing the specified document or folder is specified in the FLR parameter.

This is a required parameter.

Folder (FLR)

Specifies the folder path that contains the document or folder specified in the **Document library object (DLO)** parameter. The FLR parameter must be *NONE if the DLO parameter is *DOCID, *LADNTSP, or *SYSOBJNAM.

*NONE

The document or folder is not contained in a folder, or is identified using the **DOCID**, **LADNTSP**, or **SYSOBJNAM** parameter.

name   Specify the name of the folder that contains the document or folder named in the DLO parameter.

Document identifier (DOCID)

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

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

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

*NONE

A library-assigned document name is not specified for the object.

name   Specify the library-assigned name of the document or folder object. The library-assigned

document name can be determined by using the Query Document Library (QRYDOCLIB) command or by the message returned from the File Document (FILDOC) or other command.

LADN timestamp (LADNTSP)

Specifies the library assigned document name (LADN) timestamp of the document or folder object. A LADN timestamp is specified only if *LADNTSP is specified on the **Document library object (DLO)** parameter.

*NONE
> A LADN timestamp is not specified for the object.

timestamp
> Specify the LADN timestamp of the document library object.

System object name (SYSOBJNAM)

Specifies the system object name of the document or folder object. A system object name is specified only if *SYSOBJNAM is specified on the **Document library object (DLO)** parameter.

*NONE
> A system object name is not specified for the object.

name   Specify the system object name of the document library object.

Object class (OBJCLS)

Specifies the class of the object to locate.

*DOC   The specified DLO is a filed document.

*FLR   The specified DLO is a folder.

*DST   The specified DLO is a distribution document.

CL var for RTNDLO (12) (RTNDLO)

Specifies the name of a 12-character CL variable used to retrieve the user-assigned name of the selected document or folder. The *NONE value is returned for a distribution document or a document without a folder.

CL var for RTNFLR (63) (RTNFLR)

Specifies the name of a 63-character CL variable used to retrieve the folder path of the selected document or folder. The *NONE value is returned for a distribution document, a document without a folder, or a first-level folder.

CL var for RTNDOCID (24) (RTNDOCID)

Specifies the name of a 24-character CL variable used to retrieve the library-assigned document name of the selected object. Refer to the **Document identifier (DOCID)** parameter for the format of the library-assigned document name.

CL var for RTNLADNTSP (16) (RTNLADNTSP)

Specifies the name of a 16-character CL variable used to retrieve the timestamp from the LADN of the selected object. The variable is in the form YYYYMMDDHHMNSSNN.

CL var for RTNOBJNAM (10) (RTNOBJNAM)

Specifies the name of a 10-character CL variable used to retrieve the system object name of the selected object.

CL var for RTNOBJCLS (8) (RTNOBJCLS)

Specifies the name of a 8-character CL variable used to retrieve the object class. A value of *DOC is returned for a filed document, *FLR for a folder, and *DST for a distribution document.

CL var for RTNASP (2 0) (RTNASP)

Specifies the name of a variable used to return the auxiliary storage pool ID. In control language (CL) programs, this should be a decimal variable of length (2 0). The following values can be returned:

1   The object is in the system auxiliary storage pool.

2-32   The object is in a user auxiliary storage pool.

CL var for RTNOVRFLW (1) (RTNOVRFLW)

Specifies the name of a 1-character CL variable used to retrieve the overflow status of the object, where:

- N = No, the object has not overflowed its ASP
- Y = Yes, the object has overflowed its ASP and part or all of the object resides in the system ASP

Examples

```
RTVDLONAM  DLO(MYDOC) FLR(MYFLR) OBJCLS(*DOC) +
           RTNDOCID(&DOCID)
```

This command finds the document MYDOC in folder MYFLR and returns its document identifier in the variable &DOCID.

Error messages

*ESCAPE Messages

CPF8AC0
&1 command failed.

CPF8AC1
Not authorized to distribution documents.

CPF8A75
Not authorized to access folder &1.

CPF8A77
Folder &1 not found.

CPF8A82
Document &2 not found in folder &1.

CPF8A83
Not authorized to access document &2 in folder &1.

Retrieve Document (RTVDOC)

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

The Retrieve Document (RTVDOC) command allows you to retrieve information from a specific document.

Restrictions:

- To retrieve any records from the document to a database file, you must have use (*USE) authority to the document or be working on behalf of a user that has *USE authority to the document.
- To check out the document, you must have at least change (*CHANGE) authority to the document, or be working on behalf of a user that has *CHANGE authority to the document.
- To work on behalf of another user, you must have either all object (*ALLOBJ) special authority or special permission (granted with the Grant User Permission (GRTUSRPMN) command).

Parameters

Keyword	Description	Choices	Notes
FROMDOC	From document	*Character value*, *DOCID	Required, Positional 1
FROMFLR	From folder	*Character value*, ***NONE**	Optional, Positional 2
OUTFILE	File to receive output	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	
OUTDTATYP	Type of data for output	Single values: ***DFT**, *ALL Other values (up to 21 repetitions): *DOCD, *DOCCLS, *SUBJECT, *AUTHOR, *FILCAB, *CPYLST, *KWD, *DOCDATE, *FILDATE, *CRTDATE, *CHGDATE, *EXPDATE, *ACTDATE, *CMPDATE, *REF, *STATUS, *PROJECT, *IDP, *DOC	Optional
USRID	User identifier	Single values: ***CURRENT** Other values: *Element list*	Optional
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	
DOCID	Document identifier	*Character value*, ***NONE**	Optional
CHKOUT	Check out	***NO**, *YES	Optional

Keyword	Description	Choices	Notes
CMDCHRID	Command character identifier	Single values: ***SYSVAL**, *DEVD Other values: *Element list*	Optional
	Element 1: Graphic character set	*Integer*	
	Element 2: Code page	*Integer*	

From document (FROMDOC)

Specifies the name of the document being retrieved.

name   Specify the user-assigned name of the document being retrieved.

*DOCID

The document being retrieved is identified by the library-assigned document name that must be specified on the **Document identifier (DOCID)** parameter.

From folder (FROMFLR)

Specifies the name of the folder that contains the document being retrieved. This is the name assigned to the folder when it is created. This parameter must be specified if a document name is specified.

*NONE

No folder name is specified when the document is identified by the library-assigned document name.

name   Specify the name of the folder that contains the retrieved document.

File to receive output (OUTFILE)

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

Qualifier 1: File to receive output

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

Qualifier 2: Library

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

*CURLIB

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

name   Specify the name of the library to be searched.

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

This file can be reused when other RTVDOC commands are used. Output can be added to the file or can replace the existing records. The IBM-supplied database file QAOSIRTV in library QSYS cannot be specified.

Output member options (OUTMBR)

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

Element 1: Member to receive output

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

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

Element 2: Replace or add records

*REPLACE
> The system clears the existing member and adds the new records.

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

Type of data for output (OUTDTATYP)

Specifies the parts of information about the document that are written to the database file.

*DFT   The document information record is written. This is the same as specifying *DOCD and *DOC.
- Record code - 105, Document description
- Record code - 800, Document data

*ALL   All information records are written.

*ACTDATE
> The action due date record is written. The record code is 135.

*AUTHOR
> The author records are written. The record code is 145.

*CHGDATE
> The date last changed record is written. The record code is 130.

*CMPDATE
> The completion date record is written. The record code is 140.

*CPYLST
> The copy list records are written. The record code is 150.

*CRTDATE
> The create date record is written. The record code is 110.

*DOCCLS
> The document class record is written. The record code is 155.

*DOCD
> The document description record is written. The record code is 105.

*DOCDATE
> The document date record is written. The record code is 120.

*EXPDATE
> The expiration date record is written. The record code is 115.

*FILCAB
> The file cabinet reference record is written. The record code is 160.

*FILDATE
> The file date record is written. The record code is 125.

*IDP   The interchange document profile (IDP) is written. The record code is 500.

*IDXDATE
> The last indexed date record is written to the output file. OfficeVision/400 text search services
> must be installed if this value is specified.

*KWD
> The keyword records are written. The record code is 170.

*PROJECT
> The project record is written. The record code is 185.

*REF   The reference record is written. The record code is 175.

*REVDATE
> The date of the last revision to the document content is written to the output file.

*STATUS
> The status record is written. The record code is 180.

*SUBJECT
> The subject records are written. The record code is 165.

*USEDATE
> The date last used record is written. The record code is 200.

*DOC   The document data record is written. The record code is 800.

User identifier (USRID)

Specifies which user ID and user ID address should be associated with the request.

Single values

*CURRENT
> You are performing the request for yourself.

Element 1: User ID

character
> Specify another user's user ID or your user ID. You must have been given permission to work on behalf of another user or have all object (*ALLOBJ) special authority.

Element 2: Address

character
> Specify another user's address or your address. You must have been given permission to work on behalf of another user or have *ALLOBJ authority.

Document identifier (DOCID)

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

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

YYYYMMDDHHMNSSHSSNSNSNSN

where:

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

*NONE
> No library-assigned document name is required when the document is identified on the **Document (DOC)** parameter.

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

Check out (CHKOUT)

Specifies that the document being retrieved can be replaced with new or changed data. If the document is read only, then specify *NO. If the document being retrieved cannot be replaced, and *YES is specified, this will cause an error to occur.

*NO   The retrieve request only reads the data. Users requesting this function need only read (*READ) authority to the document. Public authority is *READ authority.

*YES   The document data can be updated and replaced later. Users requesting this function must have change (*CHANGE) authority. The document will be unavailable for other users to update until the replacement of this document is done. The replacement can be done by using the Replace Document (RPLDOC) command.

Command character identifier (CMDCHRID)

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

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

Single values

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

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

Element 1: Graphic character set

1-32767
>Specify the graphic character set to use.

Element 2: Code page

1-32767
>Specify the code page to use.

Top

Examples

Example 1: Copying All Information

```
RTVDOC   FROMDOC(MYDOC)  FROMFLR(PERSONAL)  USRID(*CURRENT)
         OUTFILE(*CURLIB/MYFILE)  OUTMBR(*FIRST)  MBROPT(*ADD)
         OUTDTATYP(*ALL)
```

This command copies all information about document MYDOC located in folder PERSONAL for the current user of this command. CHECKOUT(*NO) is assumed; therefore, the document data can only be read. The output is directed to the database file MYFILE in the user's current library and is added to the first member in that file.

Example 2: Copying Default Information

```
RTVDOC   FROMDOC(SECOP)  FROMFLR(PERSONAL)  USRID(MARY SYSTEM1 )
         CHKOUT(*YES)
         OUTFILE(MARLIB/SECFILE)  OUTMBR(*FIRST *ADD)
```

This command copies the default information (*DOCD and *DOC) about document SECOP located in folder PERSONAL for MARY. The document can be updated with new data and then replaced. The current user of this command must have the authority to work on behalf of MARY given by Mary by using the GRTUSRPMN command. The output is directed to the database file SECFILE in Mary's library MARLIB. The output is added to the first member of SECFILE.

Top

82   IBM Systems - iSeries: i5/OS Commands Starting with RSTSYSINF (Restore System Information)

Error messages

*ESCAPE Messages

CPF900B

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

CPF900C

   Sign on and verify of user failed.

CPF905C

   Error occurred trying to find a translation table.

CPF905F

   Retrieval of document from library failed.

CPF9096

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

CPF9860

   Error occurred during output file processing.

Top

Retrieve Disk Information (RTVDSKINF)

Where allowed to run:
- Batch job (*BATCH)
- Batch program (*BPGM)
- Batch REXX procedure (*BREXX)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (*EXEC)

Threadsafe: No

The Retrieve Disk Information (RTVDSKINF) command is used to collect disk space information. The collected information is stored in a database file in library QUSRSYS. The file name depends on the auxiliary storage pool (ASP) device for which disk space information is retrieved. If the information was retrieved from the system and basic ASPs, the collected information will be stored in file QAEZDISK. If the information was retrieved from an independent ASP device, the collected information will be stored in file QAEZnnnnn, where 'nnnnn' is the ASP number of the independent ASP. The information will be stored in a data base file member named QCURRENT.

Each time this command is run, existing information in QCURRENT is written over. To save existing information in member QCURRENT, rename file QAEZDISK or QAEZDnnnnn, or copy the member to another file.

To get the most accurate results, this command should be run at a time when there is very little system activity.

Note: Do not rename member QCURRENT within file QAEZDISK or QAEZDnnnnn. If there is more than one member in QAEZDISK or QAEZDnnnnn, the results of running this command can be unpredictable.

Parameters

Keyword	Description	Choices	Notes
ASPDEV	ASP device	*Name*, ***SYSBAS**	Optional, Positional 1

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device for which disk space information is to be retrieved.

*SYSBAS
> Disk information for the system ASP and all basic ASPs is retrieved. The system ASP has an ASP number of 1. Basic ASPs have ASP numbers from 2 through 32.

name Specify the name of the ASP device for which disk space information is to be retrieved. Independent ASP devices have ASP numbers greater than 32. You can submit multiple jobs, each performing RTVDSKINF on a different ASP device, to retrieve disk space information for multiple ASP devices in parallel.

Examples

```
RTVDSKINF   ASPDEV(*SYSBAS)
```

This command retrieves disk space information for the system and basic auxiliary storage pools (ASPs), and stores it in member QCURRENT of database file QAEZDISK. Any information in member QCURRENT is overwritten.

Error messages

*ESCAPE Messages

CPF1ED1
>   Not authorized to collect disk space information.

CPF1ED2
>   File &1 is in use and cannot be accessed.

CPF1E99
>   Unexpected error occurred.

Retrieve Data Area (RTVDTAARA)

The Retrieve Data Area (RTVDTAARA) command is used in a CL program or REXX procedure to retrieve all or part of a specified data area and copy it into a variable. RTVDTAARA does not retrieve any other attributes of the data area. Existence of the data area is not required at the time the CL program is compiled.

If the job is a group job, the data area specified may be the group data area (*GDA). This data area is automatically associated with the group, and it is inaccessible from jobs outside the group. The length of this character data area is 512 bytes. More information about group jobs is in the Work Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

A local data area (*LDA) is a character data area that is 1024 bytes in length, and it is automatically associated with the job. Another job cannot access the local data area.

If the job is a prestart job, the data area specified may be the data area that contains program initialization parameter data (*PDA). This data area is automatically associated with the prestart job and is inaccessible from other jobs. The length of this character data area is 2000 bytes. More information about prestart jobs is in the Work Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

When a data area other than a local data area, group data area, or program initialization parameter data area must be retrieved during the processing of the RTVDTAARA command, the data area is locked during the retrieval operation so that commands in other jobs cannot change or destroy it until the operation is complete. If the data area is shared with other jobs and is updated in steps involving more than one command in a job, the data area should be explicitly allocated to that job until all the steps have been performed. A data area other than a local data area, group data area, or program initialization parameter data area can be explicitly allocated with the Allocate Object (ALCOBJ) command. No allocation is necessary for a local data area, group data area, or program initialization parameter data area.

Restrictions

1. To use this command, the user must have *USE authority for the data area and *EXECUTE authority for the library where the data area is located. No specific authority is required to retrieve the value of a local data area or group data area.
2. This command is conditionally threadsafe. The following restrictions apply:
   a. Retrieving DDM data areas in a job that allows multiple threads is not threadsafe.
   b. Retrieving DDM data areas will fail when more than one thread is active in a job.

Parameters

Keyword	Description	Choices	Notes
DTAARA	Data area specification	*Element list*	Required, Positional 1
	Element 1: Data area	Single values: *LDA, *GDA, *PDA Other values: *Qualified object name*	
	Qualifier 1: Data area	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Substring specifications	Single values: ***ALL** Other values: *Element list*	
	Element 1: Substring starting position	1-2000	
	Element 2: Substring length	1-2000	
RTNVAR	CL variable for returned value	*Not restricted*	Required, Positional 2

Data area specification (DTAARA)

Specifies the name of the data area whose value is retrieved.

This is a required parameter.

Element 1: Data area

Single values

*LDA   The value of the local data area is being retrieved.

*GDA   The value of the group data area is being retrieved. This value is valid only if this job is a group job.

*PDA   The value of the program initialization parameter data area is being retrieved. This value is valid only if this is a prestart job.

      Qualifier 1: Data area

      name   Specify the name of the data area.

      Qualifier 2: Library

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

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

      name   Specify the library where the data area is located.

Element 2: Substring specifications

Single values

*ALL The entire data area is retrieved.

Element 1: Substring starting position

1-2000 Specify the starting position of the data area being retrieved.

Element 2: Substring length

1-2000 Specify the length of the data area substring being retrieved. It is not possible to retrieve data outside the data area. The combination of starting position and length must always specify positions within the data area.

CL variable for returned value (RTNVAR)

Specifies the name of the variable that receives the contents of the data area.

No type conversion is performed by the RTVDTAARA command:
- If RTNVAR is declared as TYPE(*DEC), the data area retrieved must be TYPE(*DEC).
- If RTNVAR is declared as TYPE(*CHAR), the data area retrieved must be either TYPE(*CHAR) or TYPE(*LGL).
- If RTNVAR is declared as TYPE(*LGL), the data area retrieved must be either TYPE(*LGL) or TYPE(*CHAR) with a value of either '0' or '1'.

If a retrieved character string is shorter than the length of the variable specified by the RTNVAR parameter, the value is padded on the right with blanks. The retrieved string length must be less than or equal to the variable length.

When decimal data areas are retrieved, the decimals are aligned. The value of the integer portion of the data area must fit into the integer portions of the variable. Fractional data is truncated if the fraction contains more digits than the variable.

This is a required parameter.

unrestricted-value
Specify the program variable that will receive the contents of the data area.

Examples

Assume data area DA1 has been created by the following command:
```
CRTDTAARA   DTAARA(DA1)   TYPE(*CHAR)   LEN(3)   VALUE(ABC)
```

and variable &CLVAR1 has been declared as:
```
DCL   VAR(&CLVAR1)   TYPE(*CHAR)   LEN(5)   VALUE(VWXYZ)
```

Example 1: Retrieving a Character Data Area

```
RTVDTAARA   DTAARA(DA1)  RTNVAR(&CLVAR1)
```

results in:
```
&CLVAR1 = 'ABC  '
```

Example 2: Retrieving a Subset of a Character Data Area
```
RTVDTAARA   DTAARA(DA1 (2 1))  RTNVAR(&CLVAR1)
```

results in:
```
&CLVAR1 = 'B    '
```

Example 3: Retrieving a Decimal Data Area

Assume data area DA2 has been created with the following attributes:
```
CRTDTAARA   DTAARA(DA2)  TYPE(*DEC)  LEN(5 2)  VALUE(12.39)
```

and variable &CLVAR2 has been declared as:
```
DCL   VAR(&CLVAR2)  TYPE(*DEC)  LEN(5 1)  VALUE(4567.8)
```

Running this command:
```
RTVDTAARA   DTAARA(DA2)  RTNVAR(&CLVAR2)
```

results in:
```
&CLVAR2 = 0012.3
```

Note: Fractional digits are truncated instead of rounded.

Error messages

ESCAPE Messages

CPF0811
> RTNVAR parameter has incorrect length for data area.

CPF0812
> RTNVAR parameter type not valid for data area &1.

CPF0813
> Value in data area &1 not logical value.

CPF101A
> Operation on DDM data area &1 in &2 failed.

CPF1015
> Data area &1 in &2 not found.

CPF1016
> No authority to data area &1 in &2.

CPF1021
> Library &1 not found for data area &2.

CPF1022
> No authority to library &1 data area &2.

CPF1043
> Boundary alignment for data area not valid.

CPF1044

AREA parameter not valid for data area.

CPF1045

CPYPTR parameter not valid for data area.

CPF1046

DTAARA(*GDA) not valid because job not group job.

CPF1063

Cannot allocate data area &1 in library &2.

CPF1067

Cannot allocate library &1.

CPF1072

DTAARA(*PDA) not valid because job not prestart job.

CPF1087

Substring not allowed for decimal or logical data area.

CPF1088

Starting position outside of data area.

CPF1089

Substring specified for data area not valid.

CPF180B

Function &1 not allowed.

CPF9899

Error occurred during processing of command.

Top

Retrieve Group Attributes (RTVGRPA)

Where allowed to run:
- Interactive program (*IPGM)
- Interactive REXX procedure (*IREXX)

Threadsafe: No

The Retrieve Group Attributes (RTVGRPA) command retrieves information about the group in which the job that issued the command belongs. The following attributes can be retrieved:

- The group job name of the job calling the RTVGRPA command
- A list containing information about all active jobs in the group
- A count of the number of active jobs in the group
- The name of the group message queue
- The library in which the group message queue resides
- The group job name and job number of the previously active job in the group
- A control code indicating why the currently active job in the group gained control

Parameters

Keyword	Description	Choices	Notes
GRPJOB	CL var for GRPJOB (10)	*Character value*	Optional, Positional 1
GRPJOBL	CL var for GRPJOB list (1056)	*Character value*	Optional, Positional 2
GRPJOBCNT	CL var for GRPJOBCNT (3 0)	*Decimal number*	Optional, Positional 3
MSGQ	CL var for MSGQ (10)	*Character value*	Optional
MSGQLIB	CL var for MSGQLIB (10)	*Character value*	Optional
PRVGRPJOB	CL var for PRVGRPJOB (16)	*Character value*	Optional
CTLCDE	CL var for CTLCDE (3 0)	*Decimal number*	Optional

CL var for GRPJOB (10) (GRPJOB)

Specifies the name of the CL variable that receives the group job name of the job. The variable must be a character variable with a minimum length of 10 characters. If the group job name has fewer characters than the variable allows, the value is padded on the right with blanks.

CL var for GRPJOB list (1056) (GRPJOBL)

Specifies the name of the CL variable that receives the list of jobs in the group. Each entry contains the job's group job name (10), job number (6), and the 50 characters of descriptive text. The maximum number of entries in the variable is 16. The entries are ordered by most recently active job. The variable must be a character variable with a minimum length of 1056 characters. If the group job list has fewer characters than the variable allows, the value is padded on the right with blanks.

Top

CL var for GRPJOBCNT (3 0) (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 non-blank entries in the group job list. The count includes all of the active jobs in the group. Jobs that have not completely ended (jobs that have been canceled) are not counted.

Top

CL var for MSGQ (10) (MSGQ)

Specifies the name of the CL variable that receives the group message queue name. This variable must be a character variable with a minimum length of 10 characters. If the message queue name has fewer characters than the variable allows, the value is padded on the right with blanks. If there is no message queue associated with the group, the CL variable is set to the special value of *NONE.

Top

CL var for MSGQLIB (10) (MSGQLIB)

Specifies the name of the CL variable that receives the name of the library that contains the group message queue. This variable must be a character variable with a minimum length of 10 characters. If the library name has fewer characters than the variable allows, the value is padded on the right with blanks. If there is no message queue associated with the group, the CL variable is set to blanks.

Top

CL var for PRVGRPJOB (16) (PRVGRPJOB)

Specifies the name of the CL variable that receives the group job name and job number of the previously active job in the group. The variable must be a character variable with a minimum length of 16 characters. If the group job name has fewer characters than the variable allows, the value is padded on the right with blanks. If there is no previously active job in the group, the group job name portion of the CL variable is set to the special value of *NONE, and the job number portion of the CL variable is set to blanks. The CL variable is returned in the following format:

```
Group-job-name   CHAR(10)
Job-number       CHAR(6)
```

Top

CL var for CTLCDE (3 0) (CTLCDE)

Specifies the name of the CL variable that receives information about why the active job in the group has gained control. The CL variable must be a three-position decimal variable with no decimal positions. The following control codes (and their meanings) are possible:

0   There was no previously active job (no Transfer to Group Job (TFRGRPJOB) commands have been run for this group).

10   The previously active job selected this job to be transferred to on the TFRGRPJOB command.

20   The previously active job's first group program ended normally, and this job was the most recently active job in the group.

30   The previously active job was ended by the End Group Job (ENDGRPJOB) command, and this job was selected to gain control (the **Group job to be resumed (RSMGRPJOB)** parameter specified this group job).

40   The previously active job was ended by the ENDGRPJOB command and selected a job other than this job to gain control (which was ended before it could be resumed). Since this job was the most recently active job in the group, control is passed to it.

50   The previously active job was ended by the ENDGRPJOB command, and this job was the most recently active job in the group (the RSMGRPJOB parameter specified *PRV).

60   The previously active job's first group program ended abnormally, and this job was the most recently active job in the group.

70   The previously active job was ended by the End Job (ENDJOB) command, and this job was the most recently active job in the group.

<div align="right">Top</div>

Examples

Assume jobs 030001/QUSER/WORKST01 and 030002/QUSER/WORKST01 are group jobs with group job names GROUPJ1 and GROUPJ2, respectively. Also assume that message queue QGPL/GROUPMSGQ is associated with the group. If group job GROUPJ1 has just issued the TFRGRPJOB command to transfer to group job GROUPJ2, and GROUPJ2 called the following CL program:

PGM Example

```
DCL   VAR(&GRPJOBN)   TYPE(*CHAR)   LEN(10)
DCL   VAR(&GRPJOBL)   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(&GRPJOBN)  GRPJOBL(&GRPJOBL)  +
          GRPJOBCNT(&GRPCOUNT)  MSGQ(&MSGQNAME)  +
          MSGQLIB(&MSGQLIB)  PRVGRPJOB(&PRVJOB)  +
          CTLCDE(&CTLCODE)
```

The contents of the CL variables returned are as follows:

```
&GRPJOBN: GROUPJ2
&GRPJOBL: GROUPJ2  030002  50 characters of text for
                           this group job...
          GROUPJ1  030001  50 characters of text for
                           this group job...
```

Fourteen more entries, full of blanks

```
&GRPCOUNT:      002
&MSGQNAME:      GROUPMSGQ
&MSGQLIB:       QGPL
&PRVJOB:        GROUPJ1    030001
&CTLCODE:       010
```

<div align="right">Top</div>

Error messages

*ESCAPE Messages

CPF1309
> Subsystem cannot complete the &1 command.

CPF1311
> Job is not a group job.

CPF1317
> No response from subsystem for job &3/&2/&1.

CPF1351
> Function check occurred in subsystem for job &3/&2/&1.

<div align="right">Top</div>

Retrieve Image Catalog (RTVIMGCLG)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Image Catalog (RTVIMGCLG) command is used in a CL procedure to retrieve the name of the image catalog loaded in a virtual device, or to retrieve the name of the virtual device an image catalog is loaded in.

Restrictions:

- This command is shipped with public *EXCLUDE authority.
- The following authorities are required to retrieve the name of the virtual device an image catalog is loaded in:
  1. Execute (*EXECUTE) authority to library QUSRSYS.
  2. *USE authority to the image catalog.
- The following authorities are required to retrieve the name of the image catalog loaded in a virtual device:
  1. *USE authority to the device.
- This command is valid only in a compiled CL procedure.

Parameters

Keyword	Description	Choices	Notes
IMGCLG	Image catalog	*Name*	Optional
DEV	Virtual device	*Name*	Optional
DEVLOD	CL var for DEVLOD (10)	*Character value*	Optional
IMGCLGLOD	CL var for IMGCLGLOD (10)	*Character value*	Optional

Image catalog (IMGCLG)

Specifies the image catalog whose associated virtual device name is to be retrieved.

Note: A value must be specified for this parameter or the **Virtual device (DEV)** parameter.

name   Specify the name of the image catalog.

Virtual device (DEV)

Specify the virtual device whose associated image catalog name is to be retrieved.

Note: A value must be specified for this parameter or the **Image catalog (IMGCLG)** parameter.

97

name   Specify the name of the virtual device.

CL var for DEVLOD (10) (DEVLOD)

Specifies the name of the CL variable that receives the name of the current virtual device for which the specified image catalog is loaded. You must specify a CL variable declared as TYPE(*CHAR) with a minimum length of 10 characters. A value of *NONE will be returned if the name specified for the **Image catalog (IMGCLG)** parameter is not associated with a virtual device.

CL var for IMGCLGLOD (10) (IMGCLGLOD)

Specifies the name of the CL variable that receives the name of the image catalog currently loaded for the specified virtual device. You must specify a CL variable declared as TYPE(*CHAR) with a minimum length of 10 characters. A value of *NONE will be returned if the name specified for the **Virtual device (DEV)** parameter is not associated with an image catalog.

Examples

Example 1: Retrieving the Loaded Virtual Device Name
```
DCL   VAR(&MYLODDEV)  TYPE(*CHAR)  LEN(10)
RTVIMGCLG   IMGCLG(MYCLG)  DEVLOD(&MYLODDEV)
```

This command will retrieve the name of the loaded virtual device associated with image catalog **MYCLG**.

Example 2: Retrieving the Image Catalog Loaded in a Virtual Device
```
DCL   VAR(&MYCLGLOD)  TYPE(*CHAR)  LEN(10)
RTVIMGCLG   DEV(OPTVRT01)  IMGCLGLOD(&MYCLGLOD)
```

This command will retrieve the name of the image catalog associated with virtual device **OPTVRT01**.

Error messages

*ESCAPE Messages

CPFBC45
      Image catalog &1 not found.

CPF9802
      Not authorized to object &2 in &3.

CPF9820
      Not authorized to use library &1.

CPF9825
      Not authorized to device &1.

Retrieve Job Attributes (RTVJOBA)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Conditional

Parameters
Examples
Error messages

The Retrieve Job Attributes (RTVJOBA) command is used in a CL program or REXX procedure to retrieve the values of one or more job attributes and place the values into the specified variable. The attributes are retrieved for the job in which this command is used.

The CL prompt for this command lists the minimum length for retrieved variables next to the appropriate parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:
1. This command is valid only within a CL program or a REXX procedure.
2. This command is conditionally threadsafe. Refer to **Considerations for Attribute Scope and Thread Safety** for additional information.

Considerations for Attribute Scope and Thread Safety:

This command is intended to be used to retrieve job attributes for the job in which the command is used. However, if a job attribute does not exist because the attribute is now supported at the thread level only, then the thread attribute is retrieved for the thread in which this command is used.

The Scope column shows whether the attribute is scoped to the Job or to the Thread.

The Threadsafe column indicates if the attributes are considered to be threadsafe.

Yes: - Attributes marked with this value can be retrieved safely from either the initial thread or from a secondary thread.

No: - Attributes marked with this value are not threadsafe, and should not be retrieved in a multi-threaded job.

Attribute Scope and Thread Safety Table:

```
Attribute                                             Scope   Threadsafe
--------------------------                            ------  ----------
ASP group name (ASPGRP)                               Thread  Yes
Break message handling (BRKMSG)                       Job     Yes
Character identifier control (CHRIDCTL)               Job     Yes
Coded character set ID (CCSID)                        Job     Yes
Country or region ID (CNTRYID)                        Job     Yes
Current library (CURLIB)                              Thread  Yes
Current user profile (CURUSER)                        Thread  Yes
Date format (DATFMT)                                  Job     Yes
Date separator (DATSEP)                               Job     Yes
DDM conversation handling (DDMCNV)                    Job     Yes
Decimal format (DECFMT)                               Job     Yes
Default coded character set identifier (DFTCCSID)     Job     Yes
Default wait (DFTWAIT)                                Job     Yes
Device recovery action (DEVRCYACN)                    Job     Yes
End status (ENDSTS)                                   Job     Yes
Inquiry message reply (INQMSGRPY)                     Job     Yes
Job accounting code (ACGCDE)                          Job     Yes
Job date with century (CYMDDATE)                      Job     Yes
Job date without century (DATE)                       Job     Yes
Job local date and time (DATETIME)                    Job     Yes
Job log output (LOGOUTPUT)                            Job     Yes
Job message queue full action (JOBMSGQFL)            Job     Yes
Job message queue maximum size (JOBMSGQMX)           Job     Yes
Job name (JOB)                                        Job     Yes
Job number (NBR)                                      Job     Yes
Job subtype (SUBTYPE)                                 Job     Yes
Job switches (SWS)                                    Job     Yes
Job type (TYPE)                                       Job     Yes
Language ID (LANGID)                                  Job     Yes
Logging level (LOGLVL)                                Job     Yes
Logging of CL programs (LOGCLPGM)                     Job     Yes
Logging severity (LOGSEV)                             Job     Yes
Logging text (LOGTYPE)                                Job     Yes
Output queue library name (OUTQLIB)                   Job     Yes
Output queue name (OUTQ)                              Job     Yes
Print key format (PRTKEYFMT)                          Job     Yes
Print text (PRTTXT)                                   Job     Yes
Printer device name (PRTDEV)                          Job     Yes
Program return code (RTNCDE)                          Job     No
Purge (PURGE)                                         Job     Yes
Resource affinity group (RSCAFNGRP)                   Job     Yes
Run priority (RUNPTY)                                 Job     Yes
Sort sequence (SRTSEQ)                                Job     Yes
Sort sequence library (SRTSEQLIB)                     Job     Yes
Spooled file action (SPLFACN)                         Job     Yes
Status message handling (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
Thread resource affinity (THDRSCAFN)                  Thread  Yes
Time separator (TIMSEP)                               Job     Yes
Time slice (TIMESLICE)                                Job     Yes
Time-slice end pool (TSEPOOL)                         Job     Yes
Time zone abbreviated name (TIMZONABBR)              Job     Yes
Time zone description (TIMZON)                         Job     Yes
Time zone full name (TIMZONFULL)                      Job     Yes
Time zone offset (TIMOFFSET)                          Job     Yes
User library list (USRLIBL)                           Thread  Yes
User name (USER)                                      Job     Yes
```

Parameters

Keyword	Description	Choices	Notes
JOB	CL var for JOB (10)	*Character value*	Optional, Positional 1

Keyword	Description	Choices	Notes
USER	CL var for USER (10)	*Character value*	Optional, Positional 2
NBR	CL var for NBR (6)	*Character value*	Optional, Positional 3
CURUSER	CL var for CURUSER (10)	*Character value*	Optional
TYPE	CL var for TYPE (1)	*Character value*	Optional
SUBTYPE	CL var for SUBTYPE (1)	*Character value*	Optional
SYSLIBL	CL var for SYSLIBL (165)	*Character value*	Optional
CURLIB	CL var for CURLIB (10)	*Character value*	Optional
USRLIBL	CL var for USRLIBL (2750)	*Character value*	Optional
ASPGRP	CL var for ASPGRP (10)	*Character value*	Optional
LOGLVL	CL var for LOGLVL (1)	*Character value*	Optional
LOGSEV	CL var for LOGSEV (2 0)	*Decimal number*	Optional
LOGTYPE	CL var for LOGTYPE (10)	*Character value*	Optional
LOGCLPGM	CL var for LOGCLPGM (10)	*Character value*	Optional
LOGOUTPUT	CL var for LOGOUTPUT (10)	*Character value*	Optional
JOBMSGQMX	CL var for JOBMSGQMX (2 0)	*Decimal number*	Optional
JOBMSGQFL	CL var for JOBMSGQFL (10)	*Character value*	Optional
INQMSGRPY	CL var for INQMSGRPY (10)	*Character value*	Optional
STSMSG	CL var for STSMSG (7)	*Character value*	Optional
BRKMSG	CL var for BRKMSG (7)	*Character value*	Optional
DEVRCYACN	CL var for DEVRCYACN (13)	*Character value*	Optional
RTNCDE	CL var for RTNCDE (5 0)	*Decimal number*	Optional
ENDSTS	CL var for ENDSTS (1)	*Character value*	Optional
PRTDEV	CL var for PRTDEV (10)	*Character value*	Optional
OUTQ	CL var for OUTQ (10)	*Character value*	Optional
OUTQLIB	CL var for OUTQLIB (10)	*Character value*	Optional
SPLFACN	CL var for SPLFACN (10)	*Character value*	Optional
PRTTXT	CL var for PRTTXT (30)	*Character value*	Optional
PRTKEYFMT	CL var for PRTKEYFMT (10)	*Character value*	Optional
SRTSEQ	CL var for SRTSEQ (10)	*Character value*	Optional
SRTSEQLIB	CL var for SRTSEQLIB (10)	*Character value*	Optional
LANGID	CL var for LANGID (3)	*Character value*	Optional
CNTRYID	CL var for CNTRYID (2)	*Character value*	Optional
CCSID	CL var for CCSID (5 0)	*Decimal number*	Optional
DFTCCSID	CL var for DFTCCSID (5 0)	*Decimal number*	Optional
CHRIDCTL	CL var for CHRIDCTL (10)	*Character value*	Optional
DECFMT	CL var for DECFMT (1)	*Character value*	Optional
DATFMT	CL var for DATFMT (4)	*Character value*	Optional
DATSEP	CL var for DATSEP (1)	*Character value*	Optional
TIMSEP	CL var for TIMSEP (1)	*Character value*	Optional
DATE	CL var for DATE (6)	*Character value*	Optional
CYMDDATE	CL var for CYMDDATE (7)	*Character value*	Optional
DATETIME	CL var for DATETIME (20)	*Character value*	Optional

Keyword	Description	Choices	Notes
TIMZON	CL var for TIMZON (10)	*Character value*	Optional
TIMZONABBR	CL var for TIMZONABBR (10)	*Character value*	Optional
TIMZONFULL	CL var for TIMZONFULL (50)	*Character value*	Optional
TIMOFFSET	CL var for TIMOFFSET (3 0)	*Decimal number*	Optional
ACGCDE	CL var for ACGCDE (15)	*Character value*	Optional
SWS	CL var for SWS (8)	*Character value*	Optional
RUNPTY	CL var for RUNPTY (2 0)	*Decimal number*	Optional
TIMESLICE	CL var for TIMESLICE (7 0)	*Decimal number*	Optional
PURGE	CL var for PURGE (10)	*Character value*	Optional
DFTWAIT	CL var for DFTWAIT (7 0)	*Decimal number*	Optional
SBMMSGQ	CL var for SBMMSGQ (10)	*Character value*	Optional
SBMMSGQLIB	CL var for SBMMSGQLIB (10)	*Character value*	Optional
DDMCNV	CL var for DDMCNV (5)	*Character value*	Optional
TSEPOOL	CL var for TSEPOOL (10)	*Character value*	Optional
THDRSCAFN	CL var for THDRSCAFN (20)	*Character value*	Optional
RSCAFNGRP	CL var for RSCAFNGRP (10)	*Character value*	Optional

CL var for JOB (10) (JOB)

Specifies the name of the CL variable that receives the name of the job. The variable must be a character variable with a minimum length of 10 characters.

CL var for USER (10) (USER)

Specifies the name of the CL variable that receives the name of the user profile associated with the job when the job was started. The user name is the second part of the qualified job name. The variable must be a character variable with a minimum length of 10 characters.

CL var for NBR (6) (NBR)

Specifies the name of the CL variable that receives the 6-character number assigned to the job by the system. The job number is the first part of the qualified job name (job-number/user-name/job-name).

CL var for CURUSER (10) (CURUSER)

Specifies the name of the CL variable that receives the name of the current user profile. The variable must be a character variable with a minimum length of 10 characters.

CL var for TYPE (1) (TYPE)

Specifies, the name of the CL variable that receives the 1-character value representing the environment of the job. A character value of 0 indicates that the job is running as a batch job, and a 1 indicates an interactive job. The variable must be a character variable with a minimum length of 1 character.

CL var for SUBTYPE (1) (SUBTYPE)

Specifies the name of the CL variable that receives the subtype value for the environment of the job. The variable must be a character variable with a minimum length of 1 character. The following values can be returned:

Value **Environment**

*   The job has no subtype

E   The job is running as an evoked job

T   The job is running as a Multiple Requester Terminal (MRT) job

J   The job is running as a prestart job

P   The job is running as a print driver

CL var for SYSLIBL (165) (SYSLIBL)

Specifies the name of the CL variable that receives the system portion of the thread's library list. Each library name returned is left-justified in an 11-character field and padded on the right with blanks. The variable must be a character variable with a minimum of 165 characters.

CL var for CURLIB (10) (CURLIB)

Specifies the name of a CL variable that receives the name of the current library for the thread. The variable must be a character variable with a minimum length of ten characters.

Note: If the thread does not have a current library, a value of *NONE is returned in this variable.

CL var for USRLIBL (2750) (USRLIBL)

Specifies the name of the CL variable that receives the user portion of the thread's library list. Each library name returned is left-justified in an 11-character field and padded on the right with blanks. If the CL variable is too small for the library list, an exception is signaled and no library names are returned. The variable must be a character variable with a minimum length of 275 characters. A character variable with a minimum length of 2750 characters is needed when the user portion of the library list contains 250 library names.

CL var for ASPGRP (10) (ASPGRP)

Specifies the name of a CL variable that receives the auxiliary storage pool (ASP) group name. The variable must be a character variable with a minimum length of 10 characters. The special value *NONE is returned if there is no ASP group for the thread.

CL var for LOGLVL (1) (LOGLVL)

Specifies the name of the CL variable that receives the 1-character value, ranging from 0 through 4 that is the message logging level being used to determine the type of message information logged in the job log. The variable must be a character variable with a minimum length of 1 character.

CL var for LOGSEV (2 0) (LOGSEV)

Specifies the name of the CL variable that receives the 2-digit value, ranging from 00 through 99, which is the severity level that is used in conjunction with the logging level to determine which error messages are logged in the job log. The variable must be a 2-digit decimal variable specified with no decimal positions.

CL var for LOGTYPE (10) (LOGTYPE)

Specifies the name of the CL variable that receives the special value that indicates the level of text that appears for any message that is written to the job log. The variable must be a character variable with a minimum length of 10 characters.

CL var for LOGCLPGM (10) (LOGCLPGM)

Specifies the name of the CL variable that receives the special value that indicates whether processed commands in a CL program are being logged in the job log. The variable must be a character variable with a minimum length of 10 characters.

CL var for LOGOUTPUT (10) (LOGOUTPUT)

Specifies the name of the CL variable that receives the special value that indicates how the job's job log is to be produced when the job completes. The variable must be a character variable with a minimum length of 10 characters. The special value *JOBEND, *JOBLOGSVR, or *PND is returned.

CL var for JOBMSGQMX (2 0) (JOBMSGQMX)

Specifies the name of a CL variable that receives the maximum size of the job message queue. The variable must be a 2-digit decimal variable with no decimal positions.

CL var for JOBMSGQFL (10) (JOBMSGQFL)

Specifies the action that should be taken when the job message queue is full. The variable must have a minimum length of 10 characters. The special value *NOWRAP, *WRAP, or *PRTWRAP is returned.

CL var for INQMSGRPY (10) (INQMSGRPY)

Specifies the name of the CL variable that receives the special value that indicates how inquiry messages are being handled by the job. The variable must be a character variable with a minimum length of 10 characters.

CL var for STSMSG (7) (STSMSG)

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.

CL var for BRKMSG (7) (BRKMSG)

Specifies the name of a CL variable that receives the special value that indicates the mode for break message handling that is in effect for the job. The variable must be a character variable with a minimum length of seven characters.

CL var for DEVRCYACN (13) (DEVRCYACN)

Specifies the name of the CL variable that receives the special value indicating the recovery action to take for the job when an I/O error is encountered on the *REQUESTER device for interactive jobs. The variable must be a character variable with a minimum length of 13 characters.

CL var for RTNCDE (5 0) (RTNCDE)

Specifies the name of the CL variable that receives the 5-digit decimal return code of an RPG, COBOL, DFU, or sort utility program. The return code is set by these programs before they return to the programs that call them. The return code indicates the completion status of the last program (of these types) that has completed processing within the job, as follows:

0   Normal return (RPG, COBOL, DFU, or Sort Utility)

1	LR (last record) indicator on (RPG)
2	Error - no halt indicator set (RPG, COBOL, DFU, or Sort Utility)
3	Halt indicator set on (one of the RPG indicators H1 through H9)

The CL variable must be a five-digit decimal variable with no decimal positions.

CL var for ENDSTS (1) (ENDSTS)

Specifies the name of the CL variable that receives the cancellation status. The single-character value indicates whether a controlled cancellation that affects the job is currently being performed. A value of 1 indicates that either the system, the subsystem in which the job is running, or the job itself is being canceled; a 0 indicates no controlled cancellation is being performed. The CL variable must be a character variable with a minimum length of 1 character.

CL var for PRTDEV (10) (PRTDEV)

Specifies the name of a CL variable that receives the name of the printer device. The variable must be a character variable with a minimum length of 10 characters.

CL var for OUTQ (10) (OUTQ)

Specifies the name of the CL variable that receives the name of the output queue that is used by the job for spooled output. The variable must be a character variable with a minimum length of 10 characters.

CL var for OUTQLIB (10) (OUTQLIB)

Specifies the name of the CL variable that receives the name of the library containing the output queue that is used by the job for spooled output. The variable must be a character variable with a minimum length of 10 characters.

CL var for SPLFACN (10) (SPLFACN)

Specifies the name of a CL variable that receives the special value indicating whether spooled files are kept with the job or detached when the job ends. The variable must be a character variable with a minimum length of 10 characters. The special value *KEEP or *DETACH is returned.

CL var for PRTTXT (30) (PRTTXT)

Specifies the name of the CL variable that receives the print text for the job. The variable must be a character variable with a minimum length of 30 characters.

CL var for PRTKEYFMT (10) (PRTKEYFMT)

Specifies the name of the CL variable that receives the print key format for the job. The variable must have a minimum length of 10 characters. The special value *NONE, *PRTBDR, *PRTHDR, or *PRTALL is returned.

CL var for SRTSEQ (10) (SRTSEQ)

Specifies the name of the CL variable that receives the name of the sort sequence table used for the job. The special value *LANGIDUNQ, *LANGIDSHR, or *HEX can be returned to the variable. The variable must be a character variable with a minimum length of 10 characters.

CL var for SRTSEQLIB (10) (SRTSEQLIB)

Specifies the name of the CL variable that receives the name of the library containing the sort sequence table to be used for the job. The variable must be a character variable with a minimum length of 10 characters. If SRTSEQ is *LANGIDUNQ, *LANGIDSHR, or *HEX, blanks are returned in the variable.

CL var for LANGID (3) (LANGID)

Specifies the name of the CL variable that receives the value indicating the language identifier to be used for the job. The variable must be a character variable with a minimum length of 3 characters.

CL var for CNTRYID (2) (CNTRYID)

Specifies the name of the CL variable that receives the value indicating the country or region identifier to be used for the job. The variable must be a character variable with a minimum length of 2 characters.

CL var for CCSID (5 0) (CCSID)

Specifies the name of a CL variable that receives the coded character set identifier value being used. The variable must be a 5-digit decimal variable specified with no decimal positions.

CL var for DFTCCSID (5 0) (DFTCCSID)

Specifies the name of a CL variable that receives the default coded character set identifier value being used for the job. The variable must be a 5-digit decimal variable with no decimal positions.

CL var for CHRIDCTL (10) (CHRIDCTL)

Specifies the name of a CL variable that receives the value being used as the character identifier control for the job. The variable must be a character variable with a minimum length of 10 characters.

CL var for DECFMT (1) (DECFMT)

Specifies the name of a CL variable that receives the character being used as the decimal format for the job. The variable must be a character variable with a minimum length of 1 character.

CL var for DATFMT (4) (DATFMT)

Specifies the name of a CL variable that receives the special value being used as the date format for the job. The variable must be a character variable with a minimum length of four characters.

CL var for DATSEP (1) (DATSEP)

Specifies the name of a CL variable that receives the character being used as the date separator character for the job. The variable must be a character variable with a minimum length of one character.

CL var for TIMSEP (1) (TIMSEP)

Specifies the name of a CL variable that receives the character being used as the time separator character for the job. The variable must be a character variable with a minimum length of one character.

CL var for DATE (6) (DATE)

Specifies the name of the CL variable that receives the date assigned to the job by the system when the job is started. The variable must be a character variable with a minimum length of 6 characters. The job date is returned in the job-date format.

CL var for CYMDDATE (7) (CYMDDATE)

Specifies the name of the CL variable that receives the date assigned to the job by the system when the job was started. The variable must be a character variable with a minimum length of 7 characters. The job date is returned in the format CYYMMDD, where C is the century, YY is the year, MM is the month and DD is the day.

CL var for DATETIME (20) (DATETIME)

Specifies the name of a CL variable that receives the current local date and time of the job. The variable must be a character variable with a minimum length of 20 characters. The local job date and time is returned in the format YYYYMMDDHHNNSSXXXXXX where YYYY is the year, MM is the month, DD is the day, HH is the hours, NN is the minutes, SS is the seconds, and XXXXXX is the microseconds.

CL var for TIMZON (10) (TIMZON)

Specifies the name of a CL variable that receives the name of the current time zone description used to calculate local job time. The variable must be a character variable with a minimum length of 10 characters.

CL var for TIMZONABBR (10) (TIMZONABBR)

Specifies the name of a CL variable that receives the current abbreviated, or short, name for the time zone used to calculate local job time. This value will contain either the Standard Time or Daylight Saving Time abbreviated name depending on whether or not Daylight Saving Time is in effect. The variable must be a character variable with a minimum length of 10 characters.

CL var for TIMZONFULL (50) (TIMZONFULL)

Specifies the name of a CL variable that receives the current full, or long, name for the time zone used to calculate local job time. This value will contain either the Standard Time or Daylight Saving Time full name depending on whether or not Daylight Saving Time is in effect. The variable must be a character variable with a minimum length of 50 characters.

CL var for TIMOFFSET (3 0) (TIMOFFSET)

Specifies the name of a CL variable that receives the current offset in minutes used to calculate local job time. This value has been adjusted for Daylight Saving Time if necessary. The variable must be a 3-digit decimal variable with no decimal positions.

CL var for ACGCDE (15) (ACGCDE)

Specifies the name of the CL variable that receives the accounting code for the job. The variable must be a character variable with a minimum length of 15 characters.

CL var for SWS (8) (SWS)

Specifies the name of the CL variable that receives the value of the eight job switches used by the job. The job switches are retrieved as a single 8-character value with each of the characters specifying a 1 or 0 as the value of the associated switch. The CL variable must be a character variable with a minimum length of 8 characters.

CL var for RUNPTY (2 0) (RUNPTY)

Specifies the name of the CL variable that receives the 2-digit value, ranging from 1 through 99, that is the processing priority for the job in which this command is used. This value is the highest run priority allowed for any thread within the job. Individual threads within the job may have a lower run priority. The variable must be a 2-digit decimal variable specified with no decimal positions.

CL var for TIMESLICE (7 0) (TIMESLICE)

Specifies the name of the CL variable that receives the 7-digit value, ranging from 8 through 9999999, that is the maximum number of milliseconds that a thread within this job can run when it is given processing time. The variable must be a 7-digit decimal variable specified with no decimal positions.

CL var for PURGE (10) (PURGE)

Specifies the name of the CL variable that receives the special value which indicates whether this job is eligible to be moved out of main storage and placed into auxiliary storage at the end of a time slice or when entering a long wait. For additional information on job purging, refer to this parameter description under the Change Job (CHGJOB) command. The variable must be a character variable with a minimum length of 10 characters.

CL var for DFTWAIT (7 0) (DFTWAIT)

Specifies the name of the CL variable that receives the 7-digit value, ranging from 1 through 9999999 (or -1 if the value is set to *NOMAX), that is the default for the maximum number of seconds that the system waits for a machine instruction to acquire a resource. The variable must be a 7-digit decimal variable specified with no decimal positions.

CL var for SBMMSGQ (10) (SBMMSGQ)

Specifies the name of the CL variable that receives the name of a message queue. The variable must be a character variable with a minimum length of 10 characters.

CL var for SBMMSGQLIB (10) (SBMMSGQLIB)

Specifies the name of the CL variable that receives the name of the library containing the message queue described previously. The variable must be a character variable with a minimum length of 10 characters.

CL var for DDMCNV (5) (DDMCNV)

Specifies the name of a CL variable that receives the special value that indicates the action taken for distributed data management (DDM) conversations on the job. The variable must be a character variable with a minimum length of 5 characters.

CL var for TSEPOOL (10) (TSEPOOL)

Specifies the name of the CL variable that receives the special value indicating whether interactive jobs are moved to another main storage pool when they reach the time slice end. The variable must be a character variable with a minimum length of 10 characters.

CL var for THDRSCAFN (20) (THDRSCAFN)

Specifies the name of a CL variable that receives the special value indicating whether or not secondary threads have affinity to the same group of system resources as the initial thread. The special value *NOGROUP or *GROUP is returned in the first 10 characters. The second 10 characters contain the special value indicating the level of affinity. The special value *NORMAL or *HIGH is returned. The variable must be a character variable with a minimum length of 20 characters.

CL var for RSCAFNGRP (10) (RSCAFNGRP)

Specifies the name of a CL variable that receives the special value indicating that the job has affinity to the same group of processors and memory as other jobs using the same routing entry or prestart job entry. The variable must be a character variable with a minimum length of 10 characters. The special value *NO or *YES is returned.

Examples

```
RTVJOBA   NBR(&JOBNBR)   DATE(&JOBDATE)   DFTCCSID(&DFTCSID)
```

This command retrieves the job number, job date, and default coded character set identifier for the job in which this command is run. The 6-digit job number is copied into the CL variable &JOBNBR. The job date is copied into the CL variable &JOBDATE; the values for both &JOBNBR and &JOBDATE must be 6 characters in length. The 5-digit DFTCCSID value is copied into the CL variable &DFTCSID; 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.

Error messages

ESCAPE Messages

CPF098A
> USRLIBL parameter size is too small.

CPF9899
> Error occurred during processing of command.

Retrieve Journal Entry (RTVJRNE)

The Retrieve Journal Entry (RTVJRNE) command allows you to get a particular journal entry and place the results in CL variables. The CL variables contain information, such as the sequence number of the retrieved entry, and are useful in automating certain types of recovery functions. The search for a journal entry can be restricted to an object, to a range of journal receivers, to a range of journal entries, to a journal code, to an entry type, to a job, to a program, to a user profile, or to a commit cycle identifier. Multiple limitation criteria can be specified. If more than one journal entry satisfies the search values specified, the first occurrence of a journal entry satisfying all of the specified search values is returned. If there is no journal entry satisfying the search values specified, the command ends with an escape message, and the return CL variables (RTNSEQNBRLRG, RTNSEQNBR, RTNJRNCDE, RTNENTTYP, RTNRCV, RTNRCVLIB and RTNJRNE) remain the same.

The order of the search through the journal entries can be ascending or descending. The search order is determined by the value specified in the SEARCH parameter. The value for the FROM parameter must come before the value specified for the TO parameter in the specified search order.

The CL prompt for this command lists the minimum length for retrieved variables next to the correct parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length, and the second number indicates the minimum number of decimal positions.

Restrictions:

- If the sequence number is reset in the range of the receivers specified, the first occurrence of FROMENTLRG or FROMENT is used, if they are specified. If TOENTLRG or TOENT is specified, the first occurance after the FROMENTLRG or FROMENT entry is used, if FROMENTLRG or FROMENT is specified. Otherwise the first occurance is used.
- The FILE, OBJ, OBJPATH, OBJFID, SUBTREE, PATTERN, JRNCDE, ENTTYP, JOB, PGM, USRPRF, CCIDLRG, CMTCYCID, and DEPENT parameters can be used to specify a subset of all available entries within a range of journal entries.
  - If no values are specified using these parameters, all available journal entries are retrieved.
  - If more than one of these parameters are specified, then a journal entry must satisfy all of the values specified on these parameters, except when *IGNFILSLT or *IGNOBJSLT is specified on the JRNCDE parameter.
  - If a journal code is specified on the JRNCDE parameter and *IGNFILSLT is the second element of that journal code, then journal entries with the specified journal code are selected if they satisfy all selection criteria except what is specified on the FILE parameter.
  - If a journal code is specified on the JRNCDE parameter and *IGNOBJSLT is the second element of that journal code, then journal entries with the specified journal code are selected if they satisfy all selection criteria except what is specified on the OBJ, OBJPATH, OBJFID, SUBTREE, and PATTERN parameters.
- The JOB, PGM, and USRPRF parameters cannot be used to specify selection criteria if one or more journal receivers in the specified receiver range was attached to the journal when a receiver size option (RCVSIZOPT) or a fixed length data option (FIXLENDTA) that would have omitted this data was in effect.
- If more than the maximum number of objects is identified (32767 objects), an error occurs and no entries are retrieved. This restriction is ignored if *ALLFILE is specified or no objects are specified.

113

• When journal caching is being used, entries that are in the cache are not retrievable.

Parameters

Keyword	Description	Choices	Notes
JRN	Journal	*Qualified object name*	Required, Positional 1
	Qualifier 1: Journal	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
FILE	Journaled physical file	Single values: *ALLFILE Other values (up to 300 repetitions): *Element list*	Optional, Positional 2
	Element 1: File	*Qualified object name*	
	Qualifier 1: File	*Name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Member	*Name*, ***FIRST**, *ALL, *NONE	
OBJ	Objects	Values (up to 300 repetitions): *Element list*	Optional
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Object type	*FILE, *DTAARA, *DTAQ	
	Element 3: Member, if data base file	*Name*, ***FIRST**, *ALL, *NONE	
OBJPATH	Objects	Values (up to 300 repetitions): *Element list*	Optional
	Element 1: Name	*Path name*	
	Element 2: Include or omit	***INCLUDE**, *OMIT	
OBJFID	File identifier	Values (up to 300 repetitions): *Hexadecimal value*	Optional
SUBTREE	Directory subtree	***NONE**, *ALL	Optional
RCVRNG	Range of journal receivers	Single values: ***CURRENT**, *CURCHAIN Other values: *Element list*	Optional, Positional 3
	Element 1: Starting journal receiver	*Qualified object name*	
	Qualifier 1: Starting journal receiver	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Ending journal receiver	Single values: ***CURRENT** Other values: *Qualified object name*	
	Qualifier 1: Ending journal receiver	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
PATTERN	Name pattern	Values (up to 20 repetitions): *Element list*	Optional
	Element 1: Pattern	*Character value*, *****	
	Element 2: Include or omit	***INCLUDE**, *OMIT	
FROMENTLRG	Starting large sequence number	*Character value*, ***FIRST**, *LAST	Optional
FROMTIME	Starting date and time	*Element list*	Optional
	Element 1: Starting date	*Date*	
	Element 2: Starting time	*Time*	
TOENTLRG	Ending large sequence number	*Character value*, ***LAST**, *FIRST	Optional

Keyword	Description	Choices	Notes
TOTIME	Ending date and time	*Element list*	Optional
	Element 1: Ending date	*Date*	
	Element 2: Ending time	*Time*	
SEARCH	Search	***ASCEND**, *DESCEND	Optional
JRNCDE	Journal codes	Single values: ***ALL**, *CTL Other values (up to 16 repetitions): *Element list*	Optional
	Element 1: Journal code value	A, B, C, D, E, F, J, L, M, P, Q, R, S, T, U	
	Element 2: Journal code selection	***ALLSLT**, *IGNFILSLT, *IGNOBJSLT	
ENTTYP	Journal entry types	Single values: ***ALL**, *RCD Other values (up to 300 repetitions): *Character value*	Optional
JOB	Job name	Single values: ***ALL**, * Other values: *Qualified job name*	Optional
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
PGM	Program	*Name*, ***ALL**	Optional
USRPRF	User profile	*Name*, ***ALL**	Optional
CCIDLRG	Commit cycle large identifier	*Character value*, ***ALL**	Optional
DEPENT	Dependent entries	***ALL**, *NONE	Optional
ENTFMT	Entry format	***TYPE1**, *TYPE2, *TYPE3, *TYPE4, *TYPE5	Optional
FMTMINDTA	Format minimized data	***NO**, *YES	Optional
NULLINDLEN	Null value indicators length	1-8000	Optional
INCENT	Include entries	***CONFIRMED**, *ALL	Optional
FROMENT	Starting sequence number	1-9999999999, ***FIRST**, *LAST	Optional
TOENT	Ending sequence number	1-9999999999, ***LAST**, *FIRST	Optional
CMTCYCID	Commit cycle identifier	1-9999999999, ***ALL**	Optional
RTNSEQLRG	CL var for RTNSEQLRG (20)	*Character value*	Optional
RTNJRNCDE	CL var for RTNJRNCDE (1)	*Character value*	Optional
RTNENTTYP	CL var for RTNENTTYP (2)	*Character value*	Optional
RTNRCV	CL var for RTNRCV (10)	*Character value*	Optional
RTNRCVLIB	CL var for RTNRCVLIB (10)	*Character value*	Optional
RTNJRNE	CL var for RTNJRNE (1)	*Character value*	Optional
RTNSEQNBR	CL var for RTNSEQNBR (100)	*Decimal number*	Optional

Journal (JRN)

Specifies the journal from which the journal entry is retrieved.

This is a required parameter.

Qualifier 1: Journal

journal-name

      Specify the name of the journal.

Qualifier 2: Library

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

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

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

Journaled physical file (FILE)

Specifies a maximum of 300 files whose journal entries are retrieved. This parameter also specifies the file member whose journal entries are to be retrieved.

To determine which journal entries are to be retrieved, based on the specified file member name, the following is done:
- If the journal is a local journal, and if the specified file member currently exists on the system, the journal identifier id determined from the specified file member. All journal entries in the specified receiver range for that journal identifier are retrieved.
- If the journal is a remote journal, or if the specified file member does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified file member. All journal entries in the specified receiver range for those journal identifiers are retrieved. Specify the library name or *CURLIB to have entries returned for the file.

  There may be more than one journal identifier associated with a specified object within the specified receiver range. This can happen when a journaled object is deleted, and then a new object is created with the same name and journaled to the same journal.

Notes:
1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved, or restored. See the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information.
2. When specifying a database file on this parameter, journal entries with the following journal code values are retrieved only if they satisfy the values specified on the other parameters:
   - Journal code D (database file-level information entries).
   - Journal code F (file member-level information entries).
   - Journal code R (record-level information entries).
   - Journal code U (user-generated entries).
   - Other journal codes, if *IGNFILSLT is specified on that journal code. If *ALLSLT is specified on that journal code, no journal entries with that code are retrieved.

Single values

*ALLFILE
> The search for the entry being retrieved is not limited to a specified file name. All journal entries are retrieved, regardless of which objects, if any, the entries are associated with.

Element 1: File

Qualifier 1: File

*ALL   Journal entries for all physical 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 database physical file for which a journal entry is retrieved.

Qualifier 2: Library

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

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

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

Element 2: Member

*FIRST
> Journal entries for the database file and the first member in the file are retrieved.

*ALL   Journal entries for the database file and all the currently existing members in the file are retrieved.

*NONE
> Only entries for the database file are retrieved. Entries for members of the file are not retrieved.

member-name
> Specify the name of the member for which an entries are retrieved.

If *ALL is specified for the file-name element, this member name is used for all applicable files in the library. For example, if library-name/*ALL *FIRST is specified on the FILE parameter, the journal entries of the first members of all applicable files in the specified library are retrieved.

Objects (OBJ)

Specifies a maximum of 300 qualified object names whose journal entries are to be retreived The possible object types are *FILE, *DTAARA, and *DTAQ. If *FILE is specified, this parameter also specifies the name of the file member whose journal entries are to be retrieved.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, or OBJFID) may be specified, but not both.

To determine which journal entries are to be retrieved, based on the specified object name, the following is done:

- If the journal is a local journal, and if the specified object currently exists on the system, the journal identifier is determined from the specified object. All journal entries in the specified receiver range for that journal identifier are retrieved.
- If the journal is a remote journal, or if the specified object does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified object. All journal entries in the specified receiver range for those journal identifiers are retrieved. Specify the library name or *CURLIB to have entries returned for an object.

There may be more than one journal identifier associated with a specified object within the specified receiver range. This can happen when a journaled object is deleted, and then a new object is created with the same name and journaled to the same journal.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information.

2. When specifying an object on this parameter, journal entries with the following journal code values are retrieved only if they satisfy the values specified on the other parameters in addition to the object name specification:

   - Journal code D (database file-level information entries).
   - Journal code E (data area information entries).
   - Journal code F (file member-level information entries).
   - Journal code Q (data queue information entries).
   - Journal code R (record-level information entries).
   - Journal code U (user-generated entries).
   - Other journal codes, if *IGNOBJSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are retrieved.

Element 1: Object

Qualifier 1: Object

*ALL Journal entries for all objects of the specified object type in the specified library (the library name must be specified) whose journaled changes are currently in the journal receiver are retrieved. The library name must be specified. If *ALL is specified and the user does not have the required authority for all objects in the library, a message is sent and the command ends.

object-name
    Specify the name of the object whose journaled changes are to be retrieved.

Qualifier 2: Library

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

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

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

Element 2: Object type

Specify the object type of the object whose journaled changes are to be retrieved.

*FILE Entries for database files and database file members are retrieved.

*DTAARA
    Entries for data areas are retrieved.

*DTAQ
    Entries for data queues are retrieved.

Element 3: Member, if data base file

Specify the name of the member in the file whose journal entries are to be retrieved. If *ALL is specified for the first part of this parameter, the value specified for the member name is used for all applicable files in the library. For example, if *FIRST is specified, the journal entries of the first member of all applicable files in the specified library are retrieved.

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

***FIRST**
Journal entries for the database file and the first member in the file are retrieved.

***ALL** Journal entries for the database file and all the currently existing members in the file are retrieved.

***NONE**
Only entries for the database file are retrieved. Entries for members of the file are not retrieved.

member-name
Specify the name of the member for which an entries are retrieved.

If *ALL is specified for the object-name element, this member name is used for all applicable files in the library. For example, if library-name/*ALL *FILE *FIRST is specified on the OBJ parameter, the journal entries of the first members of all applicable files in the specified library are retrieved.

Objects (OBJPATH)

Specifies a maximum of 300 objects whose journal entries are to be retrieved. Only objects whose path name identifies an object of type *STMF, *DIR or *SYMLNK that are in the "root" ('/'), QOpenSys, and user-defined file systems are supported. All other objects are ignored.

This parameter is not valid for remote journals.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, or OBJFID) may be specified, but not both.

Only objects that are currently linked with the specified path name and have a journal identifier associated with them are used in journal entry selection. If the specified object does exist, the journal identifier associated with that link is used for journal entry selection. If a specified object does not exist or does not have a journal identifier associated with it, that link is not used in selecting journal entries and no error is sent.

Notes:
1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information.
2. When specifying an object on this parameter, journal entries with the following journal code values are retrieved only if they satisfy the values specified on the other parameters in addition to the object name specification:
   • Journal code B (integrated file system information entries).
   • Journal code U (user-generated entries).
   • Other journal codes, if *IGNOBJSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are retrieved.

Element 1: Name

path-name
> Entries for objects identified by the path name are retrieved.
>
> A pattern can be specified in the last part of the path name. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes. Symbolic links within the path name will not be followed. If the path name begins with the tilde character, then the path is assumed to be relative to the appropriate home directory.
>
> Additional information about path name patterns is in the Integrated file system information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Element 2: Include or omit

The second element specifies whether names that match the path name should be included or omitted from the operation. Note that in determining whether a name matches a pattern, relative name patterns are always treated as relative to the current working directory.

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

*INCLUDE
> The objects that match the object name pattern are to be included in determining what journal entries are retrieved, unless overridden by an *OMIT specification.

*OMIT
> The objects that match the object name pattern are not to be included in determining what journal entries are retrieved. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

File identifier (OBJFID)

Specifies a maximum of 300 file identifiers (FID) whose journal entries are to be retrieved. FIDs are a unique identifier associated with integrated file system related objects. This field is input in hexadecimal format. Only objects whose FID identifies an object of type *STMF, *DIR or *SYMLNK that are in the "root" ('/'), QOpenSys, and user-defined file systems are supported. All other objects are ignored.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, or OBJFID) may be specified, but not both.

To determine which journal entries are to be received, based on the specified file identifier, the following is done:

- If the journal is a local journal, and if the specified object currently exists on the system, the journal identifier is determined from the specified object. All journal entries in the specified receiver range for that journal identifier are retrieved.
- If the journal is a remote journal, or if the specified object does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified object. All journal entries in the specified receiver range for those journal identifiers are retrieved.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information.

2. When specifying an object on this parameter, journal entries with the following journal code values are retrieved only if they satisfy the values specified on the other parameters in addition to the object name specification:

- Journal code B (integrated file system information entries).
- Journal code U (user-generated entries).
- Other journal codes, if *IGNOBJSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are retrieved.

file-identifier
      Entries for objects identified with the FID are retrieved.

Directory subtree (SUBTREE)

Specifies whether the directory subtrees are included in determining the objects for which journal entries are to be retrieved.

Note: This parameter is only valid if one or more path names were specified on the OBJPATH parameter.

*NONE
      Only the objects that match the selection criteria are processed. The objects within selected directories are not implicitly processed.

*ALL   All objects that meet the selection criteria are processed in addition to the entire subtree of each directory that matches the selection criteria. The subtree includes all sub-directories and the objects within those sub-directories.

Range of journal receivers (RCVRNG)

Specifies the starting (first) and ending (last) journal receivers used in the search for a journal entry to be retrieved. The system starts the search with the starting journal receiver (as specified by the first value) and proceeds through the receiver chain until the ending journal receiver (as specified by the second value) is processed.

If *ASCEND is specified for the **Search** prompt (SEARCH parameter), journal receivers must be specified in the order of oldest to newest. If *DESCEND is specified for the **Search (SEARCH)** parameter, journal receivers must be specified in the order of newest to oldest.

Note: If the maximum number of receivers (2045) in the range is surpassed, an error occurs and no journal entries are retrieved.

Single values

*CURRENT
      The journal receiver that is currently attached when starting to retrieve journal entries is used.

*CURCHAIN
      The journal receiver chain that includes the journal receiver that is currently attached when starting to retrieve journal entries is used. This receiver chain does not cross a break in the chain. If there is a break in the chain, the receiver range is from the most recent break in the chain through the receiver that is attached when starting to retrieve journal entries.

Element 1: Starting journal receiver

Qualifier 1: Starting journal receiver

starting-journal-receiver-name
> Specify the name of the first journal receiver that contains journal entries to be retrieved.

Qualifier 2: Library

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

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

library-name
> Specify the library where the journal receiver is located.

Element 2: Ending journal receiver

Qualifier 1: Ending journal receiver

*CURRENT
> The journal receiver that is currently attached when starting to retrieve journal entries is used.

ending-journal-receiver
> Specify the name of the last journal receiver containing journal entries to be searched. If the end of the receiver chain is reached before a receiver of this name is found, an error message is sent and no journal entry is retrieved.

Qualifier 2: Library

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

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

library-name
> Specify the library where the journal receiver is located.

Name pattern (PATTERN)

Specifies a maximum of 20 patterns to be used to include or omit objects for which journal entries are to be retrieved.

Only the last part of the path name will be considered for the name pattern match. Path name delimiters are not allowed in the name pattern. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes.

If the Name Pattern parameter is not specified the default will be to match all patterns.

Note: This parameter is only valid if one or more path names were specified on the OBJPATH parameter.

Element 1: Pattern

'*'   All objects that match the input OBJPATH parameter are to be included.

name-pattern
> Specify the pattern to be used to include or omit objects for which journal entries are retrieved. Only the last part of the path name will be considered for the name pattern match. Path name delimiters are not allowed in the name pattern.

If the Name Pattern parameter is not specified the default will be to match all patterns.

Additional information about path name patterns is in the Integrated file system information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Element 2: Include or omit

The second element specifies whether names that match the pattern should be included or omitted from the operation. Note that in determining whether a name matches a pattern, relative name patterns are always treated as relative to the current working directory.

Note: The SUBTREE parameter specifies whether directory subtrees are included or omitted.

*INCLUDE
> The objects that match the object name pattern are included in the operation, unless overridden by an *OMIT specification.

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

Starting large sequence number (FROMENTLRG)

Specifies the first journal entry considered for retrieval.

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

*FIRST
> The first journal entry in the specified journal receiver range is the first entry considered for retrieval. If SEARCH(*DESCEND) is specified, FROMENT(*FIRST) is valid only if TOENTLRG(*FIRST) or TOENT(*FIRST) is also specified.

*LAST
> The last journal entry in the specified journal receiver range is the first entry considered for retrieval. If SEARCH(*ASCEND) is specified, FROMENT(*LAST) is valid only if TOENTLRG(*LAST) or TOENT(*LAST) is also specified.

starting-sequence-number
> The journal entry with the assigned sequence number is the first entry considered for retrieval. The possible range is 1 to 18,446,744,073,709,551,600.

Starting date and time (FROMTIME)

Specifies the date and time of the first journal entry considered for retrieval. The first journal entry found with the specified date and time or the next later journal entry is the starting point for the search.

Element 1: Starting date

starting-date
> Specify the date. The starting date and time of the first journal entry occurring at or after the specified date and time becomes the starting point for the search.

Element 2: Starting time

starting-time
> Specify the time. The starting date and time of the first journal entry occurring at or after the specified date and time becomes the starting point for the search.

> The time can be specified in 24-hour format with or without a time separator:

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

Ending large sequence number (TOENTLRG)

Specifies the last journal entry considered for retrieval.

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

*LAST
> The search continues until the last journal entry in the journal receiver range specified is processed. If SEARCH(*DESCEND) is specified, TOENT(*LAST) is valid only if FROMENTLRG(*LAST) or FROMENT(*LAST) is also specified.

*FIRST
> The search continues until the first journal entry in the journal receiver range specified is processed. If SEARCH(*ASCEND) is specified, TOENT(*FIRST) is only valid if FROMENTLRG(*LAST) or FROMENT(*FIRST) is also specified.

ending-sequence-number
> Specify the sequence number of the final journal entry considered for retrieval. The possible range is 1 to 18,446,744,073,709,551,600.

Note: The values specified for the FROMENT and TOENT parameter can be the same. For example, FROMENT(234) and TOENT(234) can be specified.

Ending date and time (TOTIME)

Specifies the date and time of the last entry considered for retrieval. The first journal entry found with the specified date and time, or the latest earlier journal entry is the ending point for the search.

Element 1: Ending date

ending-date

Specify the ending date. The ending date and time of the journal entry occurring at or before the specified date and time becomes the ending point for the search.

Element 2: Ending time

ending-time

Specify the time. The ending date and time of the journal entry occurring at or before the specified date and time becomes the ending point for the search.

The time can be specified in 24-hour format with or without a time separator:

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

Search (SEARCH)

Specifies the order in which the journal entries are searched to retrieve an entry.

*ASCEND

The journal entries are searched in ascending order (from the oldest entry to the newest entry).

*DESCEND

The journal entries are searched in descending order (from the newest entry to the oldest entry).

Journal codes (JRNCDE)

Specifies the journal codes of the journal entries being considered for retrieval.

Single values

*ALL   The search for the entry is not limited to a specified journal code.

*CTL   The journal entries considered for retrieval are those used to control the journal functions. The journal codes are **J** and **F**.

Element 1: Journal code value

journal-code

Specify the journal code to which journal entries are limited. Only journal entries with the specified journal code are retrieved.

An explanation of the journal codes that can be specified is in the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Element 2: Journal code selection

*ALLSLT
> The journal entries with the specified journal code are retrieved only if all other selection parameters are satisfied.

*IGNFILSLT
> Journal entries having the specified journal code are retrieved only if all selection parameters, except the FILE parameter, are satisfied.
>
> **Note:** This value is not valid for journal codes D, F, and R. This value is not valid if the OBJ, OBJPATH, or OBJFID parameters are specified.

*IGNOBJSLT
> Journal entries having the specified journal code are retrieved only if all selection parameters are satisfied except OBJ, OBJPATH, OBJFID, SUBTREE, and PATTERN.
>
> **Note:** This value is not valid for journal codes B, D, E, F, Q, and R. This value is not valid if the FILE parameter is specified.

Journal entry types (ENTTYP)

Specifies whether to limit the journal entries retrieved to those of a specified journal entry type.

Single values

*ALL   The search for the entry is not limited to a particular entry type.

*RCD   Only entries that have an entry type for record-level operations are retrieved. The following entry types are valid: BR, DL, DR, IL, PT, PX, UB, UP, and UR.

Other values

entry-type
> Specify the entry type that limits the search for the entry. Only journal entries that contain the specified entry type are considered for retrieval. Up to 300 valid entry types can be specified. More information on entry types is in the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Job name (JOB)

Specifies that the journal entries searched for retrieval are limited to the journal entries for the specified job.

Single values

*ALL   The search is not limited to entries for a specified job.

*   The search is limited to entries for the current job.

Other values

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

job-name
>   Specify the job name of the job.

user-name
>   Specify the user name of the job.

job-number
>   Specify the system-assigned job number.

Program (PGM)

Specifies that the journal entries considered for retrieval are limited to the journal entries created by the specified program.

*ALL   The search is not limited to entries for a specified program.

program-name
>   Specify the name of the program whose journal entries are considered for retrieval.

User profile (USRPRF)

Specifies that the journal entries considered for retrieval are limited to the journal entries for a specified user profile.

*ALL   The retrieval of journal entries is not limited to entries for a specified user profile.

user-profile-name
>   Specify the name of the user profile whose journal entries are considered for retrieval.

Commit cycle large identifier (CCIDLRG)

Specifies that the journal entries considered for retrieval are limited to the journal entries containing the specified commit cycle identifier. A commit cycle consists of all journal entries sharing the same commit cycle identifier. A journal entry's commit cycle identifier can be displayed by using the Display Journal (DSPJRN) command and entering option five.

Note: You can input a value for either the **Commit cycle identifier** field (CMTCYCID) or the **Commit cycle large identifier** field (CCIDLRG) but not for both.

*ALL   The search is not limited to entries for a specified commit cycle identifier.

commit-cycle-identifier
>   Specify the commit cycle identifier of the journal entries to be retrieved. The possible range is 1 to 18,446,744,073,709,551,600.

Dependent entries (DEPENT)

Specifies whether to retrieve the journal entries recording actions

- that occur as a result of a trigger program
- on records that are part of a referential constraint
- that will be ignored during an Apply 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.

Entry format (ENTFMT)

Specifies the format of the retrieved journal entry. For a description of what is represented by each of the fields in the journal entry, see the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Lists showing detailed information on the format of the retrieved journal entries are in the RTNJRNE parameter description.

Note: If ENTFMT(*TYPE1) or ENTFMT(*TYPE2) is not specified, the NULLINDLEN parameter must be specified.

Note: If the **Receiver size options** field (RCVSIZOPT) for the journal was specified as *MAXOPT3, the sequence number, commit cycle identifier, the count of entries applied or removed or relative record number fields can reach a maximum value of 18,446,744,073,709,551,600. The length of these fields for the ENTFMT(*TYPE1), ENTFMT(*TYPE2), ENTFMT(*TYPE3), and ENTFMT(*TYPE4) formats is defined to hold a 10 digit number. Only the *TYPE5 format has fields large enough to hold this maximum value. When returning these fields for a format other than *TYPE5, if a sequence number, commit cycle identifier, or count of entries applied or removed or relative record number larger than 10 digits is found, the field is set to -1 for that entry.

*TYPE1
> The retrieved journal entries are formatted to include the minimum information that can be specified.

*TYPE2
> The retrieved journal entries include the information returned when ENTFMT(*TYPE1) is specified, and the user profile field, which gives the name of the user who logged the retrieved journal entries, and the name of the system on which the entry was sent.

*TYPE3
> The retrieved journal entries include the information returned when ENTFMT(*TYPE2) is specified, and the null value indicators.

*TYPE4
> The retrieved journal entries include the information returned when ENTFMT(*TYPE3) is specified, the journal identifier, the physical file trigger indicator, and the referential constraint indicator.

*TYPE5
> The retrieved journal entries include the information returned when OUTFILFMT(*TYPE4) is

specified, in addition to the program library name, the program library ASP device name, the program library ASP number, the system sequence number, the logical unit of work, the transaction identifier, the thread identifier, the remote address, the address family, the remote port, the arm number, the receiver name, the receiver library name, the receiver library ASP device name and the receiver library ASP number.

Format minimized data (FMTMINDTA)

Specifies whether entry specific data which has been minimized on field boundaries will be returned in a readable format.

*NO   The journal entries which have entry specific data that has been minimized on field boundaries will not be returned in a readable format. Therefore, the entry specific data may not be viewable.

*YES   The journal entries which have entry specific data that has been minimized on field boundaries will be returned in a readable format. Therefore, the entry specific data is viewable and may be used for auditing purposes. The fields that were changed are accurately reflected. The fields that were not changed and were not recorded display default data and are indicated by a value of 'F9'X in the null value indicators field.

Null value indicators length (NULLINDLEN)

Specify the length used for the null value indicators portion of the retrieved entry. This parameter is not valid if ENTFMT(*TYPE1) or ENTFMT(*TYPE2) is specified. Valid values range from 1 to 8000 bytes. If the retrieved journal entry has fewer null value indicators than the specified field length, the trailing bytes in the null value indicators field is set to 'F0'X.

Null value indicators are present in journal entries for record level operations as follows:

1. The corresponding physical file has null capable fields.
2. The record image has been minimized in the entry specific data.

If the record image has not been minimized in the entry specific data, then there is one null value indicator per field in the physical file. Each indicator is one character long and can be either:

* 'F0'X = Corresponding field is not null.
* 'F1'X = Corresponding field is null.

If the record image has been minimized on file field boundaries in the entry specific data and FMTMINDTA(*YES) was specified on the RCVJRNE command, then there is one null value indicator per field in the physical file. Each indicator is one character long and can be either:

* 'F0'X = Corresponding field is not null.
* 'F1'X = Corresponding field is null.
* 'F9'X = Corresponding field was not changed and the default value for the field is returned.

If the record image has been minimized on file field boundaries in the entry specific data and FMTMINDTA(*NO) was specified on the RTVJRNE command, then an internal value is returned for the null value indicator.

Include entries (INCENT)

Specifies whether only the confirmed, or both the confirmed and unconfirmed, journal entries are retrieved. This parameter only applies when converting journal entries for output from a remote journal.

Confirmed entries are those journal entries which have been sent to this remote journal and the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is known.

Unconfirmed entries are those journal entries which have been sent to this remote journal, but the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is not known, or the object name information for those journal entries is not yet known to the remote journal. Unconfirmed journal entries can only exist within the attached receiver of a remote journal. This only applies if synchronous delivery mode is being used for a particular remote journal.

*CONFIRMED

Only those journal entries which have been confirmed are retrieved.

*ALL   All confirmed and unconfirmed journal entries are retrieved.

Starting sequence number (FROMENT)

Specifies the first journal entry considered for retrieval.

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

*FIRST

The first journal entry in the specified journal receiver range is the first entry considered for retrieval. If SEARCH(*DESCEND) is specified, FROMENT(*FIRST) is valid only if TOENTLRG(*FIRST) or TOENT(*FIRST) is also specified.

*LAST

The last journal entry in the specified journal receiver range is the first entry considered for retrieval. If SEARCH(*ASCEND) is specified, FROMENT(*LAST) is valid only if TOENTLRG(*LAST) or TOENT(*LAST) is also specified.

starting-sequence-number

The journal entry with the assigned sequence number is the first entry considered for retrieval. The possible range is 1 to 9,999,999,999.

Ending sequence number (TOENT)

Specifies the last journal entry considered for retrieval.

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

*LAST

The search continues until the last journal entry in the journal receiver range specified is processed. If SEARCH(*DESCEND) is specified, TOENT(*LAST) is valid only if FROMENTLRG(*LAST) or FROMENT(*LAST) is also specified.

*FIRST

The search continues until the first journal entry in the journal receiver range specified is

processed. If SEARCH(*ASCEND) is specified, TOENT(*FIRST) is only valid if
FROMENTLRG(*LAST) or FROMENT(*FIRST) is also specified.

ending-sequence-number
> Specify the sequence number of the final journal entry considered for retrieval. The possible
> range is 1 to 9,999,999,999.

Note: The values specified for the FROMENT and TOENT parameter can be the same. For example,
FROMENT(234) and TOENT(234) can be specified.

Commit cycle identifier (CMTCYCID)

Specifies that the journal entries considered for retrieval are limited to the journal entries containing the
specified commit cycle identifier. A commit cycle consists of all journal entries sharing the same commit
cycle identifier. A journal entry's commit cycle identifier can be displayed by using the Display Journal
(DSPJRN) command and entering option five.

Note: You can input a value for either the **Commit cycle identifier** field (CMTCYCID) or the **Commit
cycle large identifier** field (CCIDLRG) but not for both.

*ALL   The search is not limited to entries for a specified commit cycle identifier.

commit-cycle-identifier
> Specify the commit cycle identifier of the journal entries to be retrieved. The possible range is 1 to
> 9,999,999,999.

CL var for RTNSEQLRG (20) (RTNSEQLRG)

Specifies the name of the program CL character variable into which the journal entry sequence number of
the retrieved journal entry is copied. If a CL variable name is not specified, the journal entry sequence
number is not copied into the program. The specified variable must be a character variable that has a
length of twenty positions. If the retrieved sequence number is shorter than the length of the field, the
number is padded on the right with blanks.

CL var for RTNJRNCDE (1) (RTNJRNCDE)

Specifies the name of the program CL character variable into which the journal code of the retrieved
journal entry is copied. If a CL variable name is not specified, the journal code of the retrieved journal
entry is not copied into the program. The specified variable must be a character variable with a minimum
length of 1 character. If the length of the variable is longer than 1 character, it is padded on the right with
blanks.

CL var for RTNENTTYP (2) (RTNENTTYP)

Specifies the name of the program CL character variable into which the entry type of the retrieved journal entry is copied. If a CL variable name is not specified, the entry type of the retrieved journal entry is not copied into the program. The specified variable must be a character variable with a minimum length of 2 characters. If the length of the variable is longer than 2 characters, it is padded on the right with blanks.

CL var for RTNRCV (10) (RTNRCV)

Specifies the name of the program CL character variable into which the journal receiver name from which the returned journal entry was retrieved is copied. If the CL variable name is not specified, the journal receiver name is not copied into the program. The specified variable must be a character variable with a minimum length of 10 characters. If the length of the variable is longer than 10 characters, it is padded on the right with blanks.

CL var for RTNRCVLIB (10) (RTNRCVLIB)

Specifies the name of the CL character variable into which the name of the library containing the receiver of the retrieved journal entry is copied. If the CL variable name is not specified, the journal receiver library name is not copied into the program. The specified variable must be a character variable with a minimum length of 10 characters. If the length of the variable is longer than 10 characters, it is padded on the right with blanks.

CL var for RTNJRNE (1) (RTNJRNE)

Specifies the name of the program CL character variable into which the retrieved journal entry is copied. If a CL variable name is not specified, the retrieved journal entry is not copied into the program. The specified variable must be a character variable. If the retrieved journal entry is longer than the variable's field length, the entry is truncated. If the entry is shorter, it is padded on the right with blanks.

The following lists show detailed information on the format of the retrieved journal entries.

The journal entry can be retrieved in one of the following possible formats:

If ENTFMT(*TYPE1) is specified, then the format of the fields in the retrieved entry is as follows:

```
Field Name            Field Attributes
--------------------  ---------------------------
ENTRY LENGTH          TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER (1)   TYPE(*DEC) LEN(10 0)
JOURNAL CODE          TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE    TYPE(*CHAR) LEN(2)
DATE                  TYPE(*CHAR) LEN(6)
TIME                  TYPE(*DEC) LEN(6 0)
JOB NAME              TYPE(*CHAR) LEN(10)
USER NAME             TYPE(*CHAR) LEN(10)
JOB NUMBER            TYPE(*DEC) LEN(6 0)
PROGRAM NAME          TYPE(*CHAR) LEN(10)
OBJECT NAME           TYPE(*CHAR) LEN(10)
OBJECT LIBRARY        TYPE(*CHAR) LEN(10)
MEMBER NAME           TYPE(*CHAR) LEN(10)
COUNT/RRN (2)         TYPE(*DEC) LEN(10 0)
FLAG                  TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID (3)   TYPE(*DEC) LEN(10 0)
INCOMPLETE DATA       TYPE(*CHAR) LEN(1)
MINIMIZED ENTRY DATA  TYPE(*CHAR) LEN(1)
RESERVED              TYPE(*CHAR) LEN(6)
ENTRY-SPECIFIC DATA   TYPE(*CHAR) LEN(up to 32642)
Notes:
(1)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the sequence number is larger than 10
digits.
(2)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the count of entries applied or removed
or relative record number is larger than 10 digits.
(3)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the commit cycle identifier is larger
than 10 digits.
```

If ENTFMT(*TYPE2) is specified, then the format of the fields in the retrieved entry is as follows:

```
Field Name            Field Attributes
--------------------  ---------------------------
ENTRY LENGTH          TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER (1)   TYPE(*DEC) LEN(10 0)
JOURNAL CODE          TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE    TYPE(*CHAR) LEN(2)
DATE                  TYPE(*CHAR) LEN(6)
TIME                  TYPE(*DEC) LEN(6 0)
JOB NAME              TYPE(*CHAR) LEN(10)
USER NAME             TYPE(*CHAR) LEN(10)
JOB NUMBER            TYPE(*DEC) LEN(6 0)
PROGRAM NAME          TYPE(*CHAR) LEN(10)
OBJECT NAME           TYPE(*CHAR) LEN(10)
OBJECT LIBRARY        TYPE(*CHAR) LEN(10)
MEMBER NAME           TYPE(*CHAR) LEN(10)
COUNT/RRN (2)         TYPE(*DEC) LEN(10 0)
FLAG                  TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID (3)   TYPE(*DEC) LEN(10 0)
USER PROFILE          TYPE(*CHAR) LEN(10)
SYSTEM NAME           TYPE(*CHAR) LEN(8)
INCOMPLETE DATA       TYPE(*CHAR) LEN(1)
MINIMIZED ENTRY DATA  TYPE(*CHAR) LEN(1)
RESERVED              TYPE(*CHAR) LEN(18)
ENTRY-SPECIFIC DATA   TYPE(*CHAR) LEN(up to 32612)
Notes:
(1)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the sequence number is larger than 10
digits.
(2)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the count of entries applied or removed
or relative record number is larger than 10 digits.
(3)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the commit cycle identifier is larger
than 10 digits.
```

Note: If ENTFMT(*TYPE3) is specified, the following information is not available in this format:

- Incomplete Data indicating if the journal entry data is incomplete due to either LOB fields or Byte Stream File operations.
- Minimized Entry Specific Data indicating if the journal entry has minimized entry specific data because the journal had MINENTDTA specified for the object type of the journal entry.

See the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information on the incomplete data indicator, the minimized entry specific data indicator, and these journal entries.

If ENTFMT(*TYPE3) is specified and a value is specified on the NULLINDLEN parameter, the format of the retrieved journal entry is as follows:

```
Field Name             Field Attributes
--------------------   ---------------------------
ENTRY LENGTH           TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER (3)    TYPE(*DEC) LEN(10 0)
JOURNAL CODE           TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE     TYPE(*CHAR) LEN(2)
TIMESTAMP              TYPE(*TIMESTAMP) LEN(26)
JOB NAME               TYPE(*CHAR) LEN(10)
USER NAME              TYPE(*CHAR) LEN(10)
JOB NUMBER             TYPE(*DEC) LEN(6 0)
PROGRAM NAME           TYPE(*CHAR) LEN(10)
OBJECT NAME            TYPE(*CHAR) LEN(10)
OBJECT LIBRARY         TYPE(*CHAR) LEN(10)
MEMBER NAME            TYPE(*CHAR) LEN(10)
COUNT/RRN (4)          TYPE(*DEC) LEN(10 0)
FLAG                   TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID (5)    TYPE(*DEC) LEN(10)
USER PROFILE           TYPE(*CHAR) LEN(10)
SYSTEM NAME            TYPE(*CHAR) LEN(8)
NULL VALUE INDICATORS  TYPE(*CHAR) field-length (1)
ENTRY-SPECIFIC DATA    TYPE(*CHAR) ((up to 32618)
                           minus (field length))  (2)
Notes:
(1) The length of this field is the length
specified on the NULLINDLEN parameter.
(2) The length of this portion of the entry
depends on the length specified on the RTNJRNE
parameter and the length specified on the
NULLINDLEN parameter.
(3)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the sequence number is larger than 10
digits.
(4)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the count of entries applied or removed
or relative record number is larger than 10 digits.
(5)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the commit cycle identifier is larger
than 10 digits.
```

If ENTFMT(*TYPE4) is specified and a value is specified on the NULLINDLEN parameter, the format of the retrieved journal entry is as follows:

```
Field Name              Field Attributes
--------------------    ---------------------------
ENTRY LENGTH            TYPE(*DEC)  LEN(5 0)
SEQUENCE NUMBER (3)     TYPE(*DEC)  LEN(10 0)
JOURNAL CODE            TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE      TYPE(*CHAR) LEN(2)
TIMESTAMP               TYPE(*TIMESTAMP) LEN(26)
JOB NAME                TYPE(*CHAR) LEN(10)
USER NAME               TYPE(*CHAR) LEN(10)
JOB NUMBER              TYPE(*DEC)  LEN(6 0)
PROGRAM NAME            TYPE(*CHAR) LEN(10)
OBJECT NAME             TYPE(*CHAR) LEN(10)
OBJECT LIBRARY          TYPE(*CHAR) LEN(10)
MEMBER NAME             TYPE(*CHAR) LEN(10)
COUNT/RRN (4)           TYPE(*DEC)  LEN(10 0)
FLAG                    TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID (5)     TYPE(*DEC)  LEN(10)
USER PROFILE            TYPE(*CHAR) LEN(10)
SYSTEM NAME             TYPE(*CHAR) LEN(8)
JOURNAL IDENTIFIER      TYPE(*CHAR) LEN(10)
REF CONSTRAINT          TYPE(*CHAR) LEN(1)
TRIGGER                 TYPE(*CHAR) LEN(1)
INCOMPLETE DATA         TYPE(*CHAR) LEN(1)
IGNORE DURING           TYPE(*CHAR) LEN(1)
APYJRNCHG OR RMVJRNCHG
MINIMIZED ENTRY DATA    TYPE(*CHAR) LEN(1)
RESERVED                TYPE(*CHAR) LEN(5)
NULL VALUE INDICATORS   TYPE(*CHAR) field-length (1)
ENTRY-SPECIFIC DATA     TYPE(*CHAR) ((up to 32598)
                          minus (field length))  (2)
Notes:
(1) The length of this field is the length
specified on the NULLINDLEN parameter.
(2) The length of this portion of the entry
depends on the length specified on the RTNJRNE
parameter and the length specified on the
NULLINDLEN parameter.
(3)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the sequence number is larger than 10
digits.
(4)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the count of entries applied or removed
or relative record number is larger than 10 digits.
(5)  When the RCVSIZOPT of the journal is *MAXOPT3, this field
will be set to -1 if the commit cycle identifier is larger
than 10 digits.
```

If ENTFMT(*TYPE5) is specified and a value is specified on the NULLINDLEN parameter, the format of the retrieved journal entry is as follows:

```
Field Name              Field Attributes
--------------------    --------------------------
ENTRY LENGTH            TYPE(*DEC)  LEN(5 0)
SEQUENCE NUMBER         TYPE(*CHAR) LEN(20)
JOURNAL CODE            TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE      TYPE(*CHAR) LEN(2)
TIMESTAMP               TYPE(*TIMESTAMP) LEN(26)
JOB NAME                TYPE(*CHAR) LEN(10)
USER NAME               TYPE(*CHAR) LEN(10)
JOB NUMBER              TYPE(*DEC)  LEN(6 0)
PROGRAM NAME            TYPE(*CHAR) LEN(10)
PROGRAM LIBRARY NAME    TYPE(*CHAR) LEN(10)
PROGRAM ASP DEVICE      TYPE(*CHAR) LEN(10)
NAME
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         TYPE(*CHAR) LEN(20)
NUMBER
RECEIVER NAME           TYPE(*CHAR) LEN(10)
RECEIVER LIBRARY NAME   TYPE(*CHAR) LEN(10)
RECEIVER ASP DEVICE     TYPE(*CHAR) LEN(10)
NAME
RECEIVER ASP NUMBER     TYPE(*DEC)  LEN(5 0)
ARM NUMBER              TYPE(*DEC)  LEN(5 0)
THREAD IDENTIFIER       TYPE(*CHAR) LEN(8)
THREAD IDENTIFIER HEX   TYPE(*CHAR) LEN(16)
ADDRESS FAMILY          TYPE(*CHAR) LEN(1)
REMOTE PORT             TYPE(*DEC)  LEN(5 0)
REMOTE ADDRESS          TYPE(*CHAR) LEN(46)
LOGICAL UNIT OF WORK    TYPE(*CHAR) LEN(39)
TRANSACTION IDENTIFIER  TYPE(*CHAR) LEN(140)
OBJECT TYPE             TYPE(*CHAR) LEN(7)
FILE TYPE INDICATOR     TYPE(*CHAR) LEN(1)
NESTED COMMIT LEVEL     TYPE(*DEC)  LEN(7 0)
RESERVED                TYPE(*CHAR) LEN(5)
NULL VALUE INDICATORS   TYPE(*CHAR) field-length (1)
ENTRY-SPECIFIC DATA     TYPE(*CHAR) ((up to 32212)
                            minus (field length))  (2)
Notes:
(1) The length of this field is the length
specified on the NULLINDLEN parameter.
(2) The length of this portion of the entry
depends on the length specified on the RTNJRNE
parameter and the length specified on the
NULLINDLEN parameter.
```

CL var for RTNSEQNBR (10 0) (RTNSEQNBR)

Specifies the name of the program CL decimal variable into which the journal entry sequence number of the retrieved journal entry is copied. If a CL variable name is not specified, the journal entry sequence number is not copied into the program. The specified variable must be a decimal variable that has a length of ten positions with no decimal positions. If the sequence number is larger than 9,999,999,999, this

value is set to -1, left-aligned and padded on the right with blanks. Use the **CL var for RTNSEQLRG (20) (RTNSEQLRG)** parameter to retrieve a value larger than 9,999,999,999.

Examples

Example 1

Assume the following variables are specified:

```
DCL  &SEQ      TYPE(*DEC)  LEN(10 0)
DCL  &JRNENT   TYPE(*CHAR) LEN(200)
DCL  &RCVNAME  TYPE(*CHAR) LEN(10)
DCL  &RCVLIB   TYPE(*CHAR) LEN(10)
```

and this command is run:

```
RTVJRNE   JRN(MYLIB/JRNA)  ENTTYP(PR)  +
            RTNSEQNBR(&SEQ)  RTNJRNE(&JRNENT)
```

Since no starting journal entry is specified in this command, the first entry from the journal receiver that is currently attached to the journal JRNA in the library MYLIB, when starting to retrieve entries, is considered for retrieval. The first entry in any receiver is always an identifier for the previously-attached receiver. This first receiver entry is known as a type PR entry, and it contains the name of the previously attached receiver in its entry-specific data. The PR entry is the first entry in ascending order in the currently attached receiver; when it is found, the entry is placed into a CL variable named &JRNENT.

Change Variable (CHGVAR) can then be used to separate the name and library of the previous journal receiver, found in the entry specific data, as follows:

```
CHGVAR   &RCVNAME  (%SST(&JRNENT 126 10))
CHGVAR   &RCVLIB   (%SST(&JRNENT 136 10))
```

Example 2

Assume the following variables are specified:

```
DCL  &ENTNO    TYPE(*DEC)  LEN(10 0)
DCL  &JCODE    TYPE(*CHAR) LEN(1)
DCL  &ETYPE    TYPE(*CHAR) LEN(2)
DCL  &RCVNAME  TYPE(*CHAR) LEN(10)
DCL  &RCVLIB   TYPE(*CHAR) LEN(10)
DCL  &JENTRY   TYPE(*CHAR) LEN(205)
```

and this command is run:

```
RTVJRNE   JRN(MYLIB/JRNLA)  OBJ(LIB1/A *FILE MBR3)  +
            RCVRNG(RCVLIB/RCV30 RCVLIB/RCV27)  ORDER(*DESCEND)  +
            JRNCDE(R)  ENTTYP(UP DL)  JOB(000666/QPGMR/PRESTRT) +
            PGM(WAKEUP)  USRPRF(MAC7)  ENTFMT(*TYPE2)  +
            RTNSEQNBR(&ENTNO)  RTNJRNCDE(&JCODE)  +
            RTNENTTYP(&ETYPE)  RTNRCV(&RCVNAME)  +
            RTNRCVLIB(&RCVLIB)  RTNJRNE(&JENTRY)
```

This command gets a journal entry, searching in descending order the journal receiver chain from receiver RCV30 in library RCVLIB to receiver RCV27 in library RCVLIB, journaled through journal JRNLA in library MYLIB, and copies the entry into the specified CL variables. The retrieved entry is an UPDATE or DELETE entry with journal code R from member MBR3 in file A in library LIB1, created in job 000666/QPGMR/PRESTRT in program WAKEUP by user profile MAC7. The retrieved journal entry includes the user profile field. The sequence number of the retrieved entry is copied into CL variable &ENTNO. The journal code of the retrieved entry is copied into CL variable &JCODE. The entry type of

the retrieved entry is copied into CL variable &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

*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
> Object not found and not journaled in specified receiver range.

CPF7061
> Conversion of journal entries failed.

CPF7062
> No entries converted or received from journal &1.

CPF7065
> Entry type (ENTTYP) not valid for journal code (JRNCDE).

CPF7072
> Retrieval of journal entry failed.

CPF7073
> No entry retrieved from journal &1 in &2.

CPF7074
> RCVRNG for specified SEARCH not valid.

CPF708D
> Journal receiver found logically damaged.

CPF709C
> JOB, PGM, and USRPRF not valid for receiver range.

CPF70A9

        OBJPATH parameter not valid for a remote journal.

CPF70AC

        File identifier &1 not found.

CPF9801

        Object &2 in library &3 not found.

CPF9802

        Not authorized to object &2 in &3.

CPF9803

        Cannot allocate object &2 in library &3.

CPF9809

        Library &1 cannot be accessed.

CPF9810

        Library &1 not found.

CPF9820

        Not authorized to use library &1.

CPF9822

        Not authorized to file &1 in library &2.

CPF9825

        Not authorized to device &1.

Top

Retrieve Library Description (RTVLIBD)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Yes

The Retrieve Library Description (RTVLIBD) command retrieves the description of a library. The values are returned (copied) to the specified variables in the program.

For parameters that are returned into CL variables by this command, the parameter descriptive title/and prompt text lists the minimum length for the CL variable. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

1. You must have some authority other than *EXCLUDE authority for the library to retrieve the attributes of the library.

2. You must have either all object (*ALLOBJ) or audit (*AUDIT) special authority to retrieve a value other than *NOTAVL for the create object auditing value.

Parameters

Keyword	Description	Choices	Notes
LIB	Library	*Name*	Required, Positional 1
TYPE	CL var for TYPE (10)	*Character value*	Optional
ASP	CL var for ASP (2 0)	*Decimal number*	Optional
ASPDEV	CL var for ASPDEV (10)	*Character value*	Optional
ASPGRP	CL var for ASPGRP (10)	*Character value*	Optional
CRTAUT	CL var for CRTAUT (10)	*Character value*	Optional
CRTOBJAUD	CL var for CRTOBJAUD (10)	*Character value*	Optional
TEXT	CL var for TEXT (50)	*Character value*	Optional

Library (LIB)

Specifies the library for which attributes are to be retrieved. If a variable is specified, it must be 10 characters in length and contain a library name.

This is a required parameter.

name   Specify the name of the library for which attributes are to be retrieved.

CL var for TYPE (10) (TYPE)

Specifies a 10-character variable used to return the library type. The value PROD or the value TEST is returned.

CL var for ASP (2 0) (ASP)

Specifies a decimal (2 0) variable used to return the number of the auxiliary storage pool (ASP) from which the system allocates storage for the library. The following values can be returned:

1   The library is in the system auxiliary storage pool.

2-32   The library is in a user auxiliary storage pool.

-1   The library is in a primary or secondary auxiliary storage pool (ASP) with an ASP number larger than 32. The ASP device name for the primary or secondary ASP can be retrieved with the ASPDEV parameter.

CL var for ASPDEV (10) (ASPDEV)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the library. One of the following special values can be returned:

*N   The name of the ASP device cannot be determined.

*SYSBAS
       The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

CL var for ASPGRP (10) (ASPGRP)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) group where storage is allocated for the library. The ASP group name is the name of the primary ASP within the ASP group. One of the following special values can be returned:

*N   The name of the ASP device cannot be determined.

*SYSBAS
       The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

CL var for CRTAUT (10) (CRTAUT)

Specifies a 10-character variable used to return the create authority value of the library. The value *SYSVAL, *CHANGE, *ALL, *USE, or *EXCLUDE, or the name of an authorization list is returned.

CL var for CRTOBJAUD (10) (CRTOBJAUD)

Specifies a 10-character variable used to return the auditing value of the library. A value of *NOTAVL will be returned if you do not have either all object (*ALLOBJ) or audit (*AUDIT) special authority. The values that can be returned include *SYSVAL, *NONE, *USRPRF, *CHANGE, *ALL, and *NOTAVL. See the **Create object auditing (CRTOBJAUD)** parameter on the Create Library (CRTLIB) command for more information.

Top

CL var for TEXT (50) (TEXT)

Specifies a 50-character CL variable used to return the text description of the library.

Top

Examples

Assume that the library named TESTLIB was created as follows:

```
CRTLIB  LIB(TESTLIB)  CRTAUT(*ALL)  TEXT('John Smith library')
```

PGM Example

```
DCL   VAR(&CRTAUT)  TYPE(*CHAR)  LEN(10)
RTVLIBD  LIB(TESTLIB)  CRTAUT(&CRTAUT)
```

The RTVLIBD command will return a value of '*ALL ' in program variable &CRTAUT.

Top

Error messages

*ESCAPE Messages

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.

Top

Retrieve Member Description (RTVMBRD)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

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.

Parameters

Keyword	Description	Choices	Notes
FILE	File	*Qualified object name*	Required, Positional 1
	Qualifier 1: File	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MBR	Member	Single values: ***FIRST**, *LAST Other values: *Element list*	Optional
	Element 1: Reference member	*Generic name, name*, *FIRSTMBR, *LASTMBR	
	Element 2: Relationship	***SAME**, *NEXT, *PRV	

Keyword	Description	Choices	Notes
RTNSYSTEM	CL var for RTNSYSTEM (4)	*Character value*	Optional
RTNLIB	CL var for RTNLIB (10)	*Character value*	Optional
RTNMBR	CL var for RTNMBR (10)	*Character value*	Optional
FILEATR	CL var for FILEATR (3)	*Character value*	Optional
FILETYPE	CL var for FILETYPE (5)	*Character value*	Optional
SRCTYPE	CL var for SRCTYPE (10)	*Character value*	Optional
SRCCHGDATE	CL var for SRCCHGDATE (13)	*Character value*	Optional
CRTDATE	CL var for CRTDATE (13)	*Character value*	Optional
EXPDATE	CL var for EXPDATE (7)	*Character value*	Optional
TEXT	CL var for TEXT (50)	*Character value*	Optional
NBRCURRCD	CL var for NBRCURRCD (10 0)	*Decimal number*	Optional
NBRDLTRCD	CL var for NBRDLTRCD (10 0)	*Decimal number*	Optional
SHARE	CL var for SHARE (4)	*Character value*	Optional
DTASPCSIZ	CL var for DTASPCSIZ (15 0)	*Not restricted*	Optional
ACCPTHSIZ	CL var for ACCPTHSIZ (12 0)	*Not restricted*	Optional
CHGDATE	CL var for CHGDATE (13)	*Character value*	Optional
SAVDATE	CL var for SAVDATE (13)	*Character value*	Optional
RSTDATE	CL var for RSTDATE (13)	*Character value*	Optional
NBRDTAMBRS	CL var for NBRDTAMBRS (2 0)	*Decimal number*	Optional
USEDATE	CL var for USEDATE (7)	*Character value*	Optional
USECOUNT	CL var for USECOUNT (5 0)	*Decimal number*	Optional
RESETDATE	CL var for RESETDATE (7)	*Character value*	Optional

File (FILE)

Specifies the name and library of the file that contains the member description that is retrieved.

This is a required parameter.

Note: You must have *USE authority to the file and *READ authority to the library before the member description information can be retrieved.

The possible library values are:

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

*CURLIB
>   The current library for the job is searched. If no current entry exists in the library list, QGPL is used.

library-name
>   Specify the library name to search.

Member (MBR)

Specifies the file member whose description is retrieved. Either a single value (*FIRST or *LAST) or a double value (reference member and relationship) can be specified.

The possible single values are:

*FIRST
> The first member in a date-ordered list is retrieved.

*LAST
> The last member in a date-ordered list is retrieved.

The possible reference members in double values are:

*FIRSTMBR
> The first member in a name ordered list is retrieved. The relationship value *SAME is required.

*LASTMBR
> The last member in a name ordered list is retrieved. The relationship value *SAME is required.

member-name
> Specify the name of the reference member. The relationship of the retrieved member to the reference member is specified on the second element of this parameter (*SAME, *NEXT, or *PRV). If a variable is specified, it must be a 10-character field that contains the name of the reference member.

generic*-member-name
> Specify the starting characters of the member name followed by an asterisk. This retrieves the first member in the name ordered list that starts with the specified characters. The relationship value is required to be *SAME.

The possible relationships in double values are:

*SAME
> The reference member is retrieved.

*NEXT
> The member immediately after the reference member in a name ordered list is retrieved.

*PRV The member immediately previous to the reference member in a name ordered list is retrieved.

CL var for RTNSYSTEM (4) (RTNSYSTEM)

Specifies the name of a variable used to retrieve the name of the system from which the file was retrieved. In CL programs, this should be a 4-character variable.

The values that can be returned are *LCL (file found on the local system) and *RMT (file found on a remote system).

CL var for RTNLIB (10) (RTNLIB)

Specifies the name of a variable used to retrieve the name of the library in which the file containing the specified file member is located. In CL programs, this should be a 10-character variable.

CL var for RTNMBR (10) (RTNMBR)

Specifies the name of a variable used to retrieve the name of the file member whose description is being retrieved. In CL programs, this should be a 10-character variable.

CL var for FILEATR (3) (FILEATR)

Specifies the name of a variable used to retrieve the file attribute. In CL programs, this should be a 3-character variable.

The values that can be returned are *PF (physical file member) and *LF (logical file member).

CL var for FILETYPE (5) (FILETYPE)

Specifies the name of a variable used to retrieve the file type. In CL programs, this should be a 5-character variable.

The values that are returned are *DATA (data file member) and *SRC (source file member).

CL var for SRCTYPE (10) (SRCTYPE)

Specifies the name of a variable used to retrieve the source file member type if this is a source file member. Blanks are returned if this is not a source file member. In CL programs, this should be a 10-character variable.

CL var for SRCCHGDATE (13) (SRCCHGDATE)

Specifies the name of a 13-character CL variable used to retrieve the century, date, and time the last source file member was changed. The format is 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-AS/400 and non-System/38 files or non-source physical file return blanks.

CL var for CRTDATE (13) (CRTDATE)

Specifies the name of a variable used to retrieve the file member creation century, date, and time. In CL programs, this should be a 13-character variable. The format is CYYMMDDHHMMSS where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, D = Day, H = Hour, M = Minutes, and S = Seconds.

CL var for EXPDATE (7) (EXPDATE)

Specifies the name of a variable used to retrieve the file member expiration century and date. In CL programs, this should be a 7-character variable. The format is CYYMMDD where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, and D = Day.

*NONE is returned if no date is available.

CL var for TEXT (50) (TEXT)

Specifies the name of a variable used to retrieve the file member text. In CL programs, this should be a 50-character variable.

CL var for NBRCURRCD (10 0) (NBRCURRCD)

Specifies the name of a variable used to retrieve the current number of nondeleted records in this file member. In CL programs, this should be a 10-position decimal variable.

If the member is a keyed logical member, the number of index entries is returned. For nonkeyed logical members, the number of nondeleted records in the based-on physical file member is returned.

For a join logical file, the number of records returned is a total of all the nondeleted records in the files being joined. This number includes those records that exist in the secondary file which do not have matching records in the primary file. If omit or select criteria is used, the number of records returned will be reduced or increased by the number of records meeting the selection criteria in the specified files.

CL var for NBRDLTRCD (10 0) (NBRDLTRCD)

Specifies the name of a variable used to retrieve the current number of deleted records in this file member. In CL programs, this should be a 10-position decimal variable. Zero (0) is returned for keyed logical files. Remote non-AS/400 and non-S/38 files return a value of 0. For nonkeyed logical files, the number of deleted records in the based-on physical file member is returned.

CL var for SHARE (4) (SHARE)

Specifies the name of a variable used to retrieve a value indicating whether the open data path (ODP) allows sharing with other programs in the same job. In CL programs, this should be a 4-character variable.

Values for ODP sharing are *YES (ODP sharing is allowed) and *NO (ODP sharing is not allowed).

Remote non-AS/400 and non-S/38 files return *NO.

Top

CL var for DTASPCSIZ (15 0) (DTASPCSIZ)

Specifies the name of a variable used to retrieve the data space size (in bytes) of this file member. In CL programs, this should be a 15-position decimal variable. Zero (0) is returned if this is a logical file member.

Top

CL var for ACCPTHSIZ (12 0) (ACCPTHSIZ)

Specifies the name of a variable used to retrieve the access path size (in bytes) for this file member. In CL programs, this should be a 12-position decimal variable. Zero (0) is returned if the file member is non-keyed. Remote non-AS/400 and non-S/38 files return a value of 0.

Top

CL var for CHGDATE (13) (CHGDATE)

Specifies the name of a variable used to retrieve the file change century, date, and time. In CL programs, this should be a 13-character variable. The format is CYYMMDDHHMMSS where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, D = Day, H = Hour, M = Minutes, and S = Seconds.

Top

CL var for SAVDATE (13) (SAVDATE)

Specifies the name of a variable used to retrieve the file member and the save century, date, and time. In CL programs, this should be a 13-character variable. The format is CYYMMDDHHMMSS where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, D = Day, H = Hour, M = Minutes, and S = Seconds.

Blanks are returned if no date is available. Remote non-AS/400 and non-System/38 files return blanks.

Top

CL var for RSTDATE (13) (RSTDATE)

Specifies the name of a variable used to retrieve the file member and restore century, date, and time. In CL programs, this should be a 13-character variable. The format is 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. Remote non-AS/400 and non-System/38 files return blanks.

CL var for NBRDTAMBRS (2 0) (NBRDTAMBRS)

Specifies the name of a variable used to retrieve the number of data file members for this logical file member. In CL programs, this should be a 2-position decimal variable. If the member is a physical file member, a value of 0 is returned.

CL var for USEDATE (7) (USEDATE)

Specifies the name of a variable used to return the member last used century and date. In CL programs, this should be a 7-character variable. The format is CYYMMDD where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, and D = Day.

CL var for USECOUNT (5 0) (USECOUNT)

Specifies the name of a variable that is used to return the number of days the member has been used. In CL programs, this should be a 5 decimal variable. If the member does not have a last used date, 0 is returned.

CL var for RESETDATE (7) (RESETDATE)

Specifies the name of a variable that is used to return the century and date the days used count was last reset to 0. In CL programs, this should be a 7-character variable. The format is 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 are returned.

Examples

Assume the user has a file named MYFILE in library MYLIB (which is the current library) with members QMEMBER, BMEMBER, ZMEMBER, and JMEMBER (created in that order).

Also assume the following variables are specified in the CL program:

```
DCL   &LIB        TYPE(*CHAR) LEN(10)
DCL   &MBR        TYPE(*CHAR) LEN(10)
DCL   &SYS        TYPE(*CHAR) LEN(4)
DCL   &MTYPE      TYPE(*CHAR) LEN(5)
DCL   &CRTDATE    TYPE(*CHAR) LEN(13)
DCL   &CHGDATE    TYPE(*CHAR) LEN(13)
DCL   &TEXT       TYPE(*CHAR) LEN(50)
DCL   &NBRRCD     TYPE(*DEC) LEN(10 0)
DCL   &SIZE       TYPE(*DEC) LEN(10 0)
```

Example 1: Retrieving Member Description Values
```
RTVMBRD   FILE(*CURLIB/MYFILE)  MBR(BMEMBER *SAME)  +
          RTNLIB(&LIB)  RTNSYSTEM(&SYS)  +
          RTNMBR(&MBR)  FILEATR(&MTYPE)  +
          CRTDATE(&CRTDATE)  TEXT(&TEXT)  +
          NBRCURRCD(&NBRRCD)  DTASPCSIZ(&SIZE)
```

This command retrieves the member description for member BMEMBER of file MYFILE located using the library list. The requested information is placed in the CL variables as follows:

- The current library name (MYLIB) is placed in the CL variable named &LIB.
- The system on which MYFILE was found is placed in the CL variable named &SYS. (*LCL means the file was found on the local system, and *RMT means the file was found on a remote system.)
- The member name (BMEMBER) is placed in the CL variable named &MBR.
- The file attribute of MYFILE is placed in the CL variable named &MTYPE. (*DATA means the member is a data member, and *SRC means the file is a source member.)
- The creation date of BMEMBER is placed in the CL variable named &CRTDATE.
- The text associated with BMEMBER is placed in the CL variable called &TEXT.
- The current number of records in BMEMBER is placed in the CL variable called &NBRRCD.
- The size of BMEMBER's data space (in bytes) is placed in the CL variable called &SIZE.

Example 1: Retrieving the Next Member Description
```
RTVMBRD   FILE(&LIB/MYFILE)  MBR(&MBR *NEXT)  +
          RTNMBR(&MBR)  CRTDATE(&CRTDATE)  +
          TEXT(&TEXT)  NBRCURRCD(&NBRRCD)  +
          DTASPCSIZ(&SIZE)
```

This command retrieves the member description for the member of file MYFILE which is "next" (in name order). The requested information is placed in the CL variables as follows:

- The next member's name after BMEMBER (JMEMBER since the file is searched in name order) in MYFILE is placed in the CL variable named &MBR.
- The creation date of JMEMBER is placed in the CL variable named &CRTDATE.
- The text associated with JMEMBER is placed in the CL variable called &TEXT.
- The current number of records in JMEMBER is placed in the CL variable called &NBRRCD.
- The size of JMEMBER's data space (in bytes) is placed in the CL variable called &SIZE.

The file can also be searched backwards. An example is:
```
RTVMBRD   FILE(*CURLIB/MYFILE)  MBR(ZMEMBER *PRV)  +
          RTNMBR(&MBR)  CHGDATE(&CHGDATE)  TEXT(&TEXT)
```

The requested information is placed in the CL variables as follows:

- The member name (QMEMBER since it is the member just previous to ZMEMBER in a name-ordered list) is placed in the CL variable named &MBR.
- The date QMEMBER was last changed is placed in the CL variable named &CHGDATE.
- The text associated with QMEMBER is placed in the CL variable called &TEXT.

If only the first part of the member name is known, you can use a generic name (or partial name) search of the list of members, as follows:

```
RTVMBRD   FILE(*LIBL/MYFILE)  MBR(JM*)  RTNMBR(&MBR)  +
          CHGDATE(&CHGDATE)  TEXT(&TEXT)
```

The requested information is placed in the CL variables as follows:

- The member name (JMEMBER since it is the first member starting with the characters JM in a name ordered list) is placed in the CL variable named &MBR.
- The date JMEMBER was last changed is placed in the CL variable named &CHGDATE.
- The text associated with JMEMBER is placed in the CL variable called &TEXT.

Error messages

*ESCAPE Messages

CPF3018
　　　Member &3 for file &1 in &2 not available.

CPF3019
　　　File &1 in library &2 has no members.

CPF3027
　　　File &1 in &2 not a database file.

CPF3038
　　　Attributes for return variable &1 not valid.

CPF3039
　　　Return variable &1 too small to hold result.

CPF3049
　　　*NEXT or *PRV member does not exist.

CPF3051
　　　File &1 in library &2 not available.

CPF325F
　　　Conversion of the text failed.

CPF327B
　　　File &1 information cannot be retrieved.

CPF8109
　　　&8 damage on physical database file &4.

CPF8110
　　　&8 damage on logical data base file &4 in &9. VLOG-&7.

CPF8111
　　　&8 damage on member &9 file &4.

CPF9803
　　　Cannot allocate object &2 in library &3.

CPF9806
　　　Cannot perform function for object &2 in library &3.

CPF9810
　　　Library &1 not found.

CPF9812

        File &1 in library &2 not found.

CPF9815

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

CPF9820

        Not authorized to use library &1.

CPF9822

        Not authorized to file &1 in library &2.

Retrieve Message (RTVMSG)

Where allowed to run: Compiled CL program or interpreted REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Yes

The Retrieve Message (RTVMSG) command is used in a CL program or REXX procedure to retrieve a specified predefined message from a message file and to copy it into CL variables. Substitution values can be specified in the MSGDTA parameter (as a single character string containing one or more concatenated message data fields) to replace the substitution variables in the predefined message text. The program can later write the message to an output device file to be printed, for example.

The CL prompt for this command lists the minimum length for retrieved variables next to the parameters that have a minimum length. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions: The user of this command must have use (*USE) authority for the message file and *USE authority for the library in which the message file is located.

Parameters

Keyword	Description	Choices	Notes
MSGID	Message identifier	*Name*	Required, Positional 1
MSGF	Message file	*Qualified object name*	Required, Positional 2
	Qualifier 1: Message file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MSGDTA	Message data field values	*Character value*	Optional
MSG	CL var for 1st level text	*Character value*	Optional
MSGLEN	CL var for MSGLEN (5 0)	*Decimal number*	Optional
SECLVL	CL var for 2nd level text	*Character value*	Optional
SECLVLLEN	CL var for SECLVLLEN (5 0)	*Decimal number*	Optional
SEV	CL var for SEV (2 0)	*Decimal number*	Optional
ALROPT	CL var for ALROPT (9)	*Character value*	Optional
LOGPRB	CL var for LOGPRB (1)	*Character value*	Optional
CCSID	Convert to CCSID	1-65535, *HEX, ***JOB**	Optional
MDTACCSID	Message data CCSID	1-65535, *HEX, ***JOB**	Optional
TXTCCSID	CL var for text CCSID (5 0)	*Decimal number*	Optional
DTACCSID	CL var for data CCSID (5 0)	*Decimal number*	Optional

Message identifier (MSGID)

Specifies the message identifier of the predefined message that is being retrieved from the specified message file.

This is a required parameter.

Message file (MSGF)

Specifies the message file that contains the predefined message to be retrieved.

This is a required parameter.

Qualifier 1: Message file

name   Specify the name of the message file containing the message to be retrieved.

Qualifier 2: Library

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

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

name   Specify the library where the message file is located.

Message data field values (MSGDTA)

Specifies the substitution values that are used in the retrieved message if the predefined message contains substitution variables. Either a character string or a CL variable containing the character string can be specified.

CL var for 1st level text (MSG)

Specifies the name of the CL character variable in the program into which the text of the retrieved message is copied. If a CL variable name is not specified, the message text is not copied into the program. This is a variable length field, but most messages are designed to be less than 132 characters long.

CL var for MSGLEN (5 0) (MSGLEN)

Specifies the name of the CL decimal variable in the program into which the total length of the message text available to be retrieved is copied.

The specified variable must be a decimal variable that has a length of five digits.

CL var for 2nd level text (SECLVL)

Specifies the name of the CL character variable in the program into which the second level message, or message help, of the retrieved message is copied. If a variable name is not specified, the message help is not copied into the program. This is a variable length field, but most message help is designed to be less than 3000 characters long.

CL var for SECLVLLEN (5 0) (SECLVLLEN)

Specifies the name of the CL decimal variable in the program into which the total length of the message help being retrieved is copied.

The specified variable must be a decimal variable that has a length of five positions.

CL var for SEV (2 0) (SEV)

Specifies the name of the CL decimal variable into which the severity code of the retrieved message is copied. The specified variable must be a decimal variable that has a length of two positions. If a variable name is not specified, the severity code of the retrieved message is not copied into the program.

CL var for ALROPT (9) (ALROPT)

Specifies the name of the CL variable into which the alert option of the retrieved message is copied. The variable must be a character variable nine positions long. If a character variable is not specified, the alert option of the retrieved message is not copied into the program.

CL var for LOGPRB (1) (LOGPRB)

Specifies whether the message will be logged in the problem log. The variable must be a character variable one position long.

Convert to CCSID (CCSID)

Specifies the coded character set identifier (CCSID) in which you want your message text returned. This applies only to text returned in the MSG and SECLVL parameters. When replacement data is substituted into the text returned in the MSG or SECLVL parameters, only the part of the replacement data that is defined as a character that can be converted (*CCHAR) is converted. The rest of the replacement data will not be converted. For more information about the *CCHAR field, see the Add Message Description (ADDMSGD) command.

*JOB   The retrieved message description is converted to the CCSID of the job before being returned.

*HEX   The retrieved message description is not converted before being returned.

coded-character-set-identifier
> Specify the CCSID that you want your message description converted to before it is returned.
>
> **Note:** The valid values range from 1 through 65535. See the Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for a list of valid CCSID values. Only CCSIDs that you can change your job to are accepted.
>
> For more information on the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Message data CCSID (MDTACCSID)

Specifies the CCSID that the supplied message data is assumed to be in. This only applies to the parts of the replacement data that are defined as *CCHAR. The rest of the replacement data will never be converted and is assumed to have a CCSID of 65535.

*JOB   The message data supplied is assumed to be in the CCSID of the job running this command.

*HEX   The message data supplied is assumed to be 65535 and is never converted.

coded-character-set-identifier
> The message data supplied is assumed to be in the CCSID specified. Valid values range from 1 through 65535. See the Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for a list of valid CCSID values.

CL var for text CCSID (5 0) (TXTCCSID)

Specifies the name of the CL variable, if any, used to return the coded character set identifier (CCSID) associated with the text returned by the MSG and SECLVL parameters. The CCSID that the message description is stored in is returned if the one of the following occurs:

- If a conversion error occurs.
- If the job has a CCSID of 65535 and you did not specify the CCSID parameter.
- If you specify *JOB for the CCSID parameter.
- If the CCSID you requested the text to be converted to is 65535.

Otherwise, the CCSID you wanted the text converted to is returned. If you do not want the text converted before it is returned to you but you do want to know the CCSID that the message description is stored in, specify 65535 for the CCSID parameter. The CCSID that the message description is stored in is returned in the TXTCCSID parameter. You can also check for a conversion error by comparing the CCSID you passed in against the TXTCCSID returned. If they are not equal and they are not 65535, a conversion error occurred.

CL var for data CCSID (5 0) (DTACCSID)

Specifies the name of the CL variable, if any, used to return the coded character set identifier (CCSID) associated with the replacement data defined as *CCHAR. All other replacement data is not converted before it is returned. The CCSID specified for the MDTACCSID parameter is returned if the one of the following occurs:

- If a conversion error occurs.

- If the job has a CCSID of 65535 and you did not specify the CCSID parameter.
- If you specify *JOB for the CCSID parameter.
- If the CCSID you requested the text to be converted to is 65535.

Otherwise the CCSID you wanted the text converted to is returned. When there is no *CCHAR replacement data in the text, 65535 is returned. You can check for a conversion error by comparing the CCSID you passed in against the DTACCSID returned. If they are not equal and they are not 65535, a conversion error occurred.

Examples

Example 1: Replacing Substitution Variables

```
RTVMSG   MSGID(UIN0145)  MSGF(INVN)  MSG(&WORK)
         MSGDTA('any old time')
```

This command retrieves the message text of the message UIN0145 stored in the INVN message file. The retrieved text is copied into the CL variable &WORK after the substitution variables are replaced with the values *any*, *old*, and *time*. This example assumes that the substitution variables &1, &2, and &3 have been defined in the message as character variables, each 4 characters long.

Example 2: Retrieving First-Level and Second-Level Message Text

```
RTVMSG   MSGID(UIN0150)  MSGF(INV)  MSG(&MSG)
         SECLVL(&SECLVL)
```

This command retrieves the first-level message text and second-level message text of the message UIN0150, which is stored in message file INV, and moves it into the CL variables &MSG and &SECLVL.

Error messages

*ESCAPE Messages

CPF2401
> Not authorized to library &1.

CPF2407
> Message file &1 in &2 not found.

CPF2411
> Not authorized to message file &1 in &2.

CPF247E
> CCSID &1 is not valid.

CPF2471
> Length of field not valid.

CPF2499
> Message identifier &1 not 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.

*STATUS Messages

CPF2419

        Message identifier &1 not found in message file &2 in &3.

*NOTIFY Messages

CPF2465

        Replacement text of message &1 in &2 in &3 not valid for format specified.

Top

Retrieve Network Attributes (RTVNETA)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Network Attributes (RTVNETA) command is used in a CL program or REXX procedure to retrieve the network attributes of the system. The values are returned (copied) to the specified variables in the program.

Restrictions: The attributes of the network attribute and the receiving variable must be compatible.

Parameters

Keyword	Description	Choices	Notes
SYSNAME	CL var for SYSNAME (8)	*Character value*	Optional
PNDSYSNAME	CL var for PNDSYSNAME (8)	*Character value*	Optional
LCLNETID	CL var for LCLNETID (8)	*Character value*	Optional
LCLCPNAME	CL var for LCLCPNAME (8)	*Character value*	Optional
LCLLOCNAME	CL var for LCLLOCNAME (8)	*Character value*	Optional
DFTMODE	CL var for DFTMODE (8)	*Character value*	Optional
NODETYPE	CL var for NODETYPE (8)	*Character value*	Optional
DTACPR	CL var for DTACPR (10 0)	*Decimal number*	Optional
DTACPRINM	CL var for DTACPRINM (10 0)	*Decimal number*	Optional
MAXINTSSN	CL var for MAXINTSSN (5 0)	*Decimal number*	Optional
RAR	CL var for RAR (5 0)	*Decimal number*	Optional
NETSERVER	CL var for NETSERVER (85)	*Character value*	Optional
ALRSTS	CL var for ALRSTS (10)	*Character value*	Optional
ALRPRIFP	CL var for ALRPRIFP (10)	*Character value*	Optional
ALRDFTFP	CL var for ALRDFTFP (10)	*Character value*	Optional
ALRLOGSTS	CL var for ALRLOGSTS (7)	*Character value*	Optional
ALRBCKFP	CL var for ALRBCKFP (16)	*Character value*	Optional
ALRRQSFP	CL var for ALRRQSFP (16)	*Character value*	Optional
ALRCTLD	CL var for ALRCTLD (10)	*Character value*	Optional
ALRHLDCNT	CL var for ALRHLDCNT (5 0)	*Decimal number*	Optional
ALRFTR	CL var for ALRFTR (10)	*Character value*	Optional
ALRFTRLIB	CL var for ALRFTRLIB (10)	*Character value*	Optional
MSGQ	CL var for MSGQ (10)	*Character value*	Optional
MSGQLIB	CL var for MSGQLIB (10)	*Character value*	Optional
OUTQ	CL var for OUTQ (10)	*Character value*	Optional
OUTQLIB	CL var for OUTQLIB (10)	*Character value*	Optional

Keyword	Description	Choices	Notes
JOBACN	CL var for JOBACN (10)	*Character value*	Optional
MAXHOP	CL var for MAXHOP (5 0)	*Decimal number*	Optional
DDMACC	CL var for DDMACC (10)	*Character value*	Optional
DDMACCLIB	CL var for DDMACCLIB (10)	*Character value*	Optional
PCSACC	CL var for PCSACC (10)	*Character value*	Optional
PCSACCLIB	CL var for PCSACCLIB (10)	*Character value*	Optional
DFTNETTYPE	CL var for DFTNETTYPE (10)	*Character value*	Optional
DFTCNNLST	CL var for DFTCNNLST (10)	*Character value*	Optional
ALWANYNET	CL var for ALWANYNET (10)	*Character value*	Optional
NWSDOMAIN	CL var for NWSDOMAIN (8)	*Character value*	Optional
ALWVRTAPPN	CL var for ALWVRTAPPN (10)	*Character value*	Optional
ALWHPRTWR	CL var for ALWHPRTWR (10)	*Character value*	Optional
VRTAUTODEV	CL var for VRTAUTODEV (5 0)	*Decimal number*	Optional
HPRPTHTMR	CL var for HPRPTHTMR (40)	*Character value*	Optional
ALWADDCLU	CL var for ALWADDCLU (10)	*Character value*	Optional
MDMCNTRYID	CL var for MDMCNTRYID (2)	*Character value*	Optional

CL var for SYSNAME (8) (SYSNAME)

Specifies the name of the CL variable that receives the current system name. The variable must be a character variable with a minimum length of 8 characters.

CL var for PNDSYSNAME (8) (PNDSYSNAME)

Specifies the name of the CL variable that receives the pending system name. The variable must be a character variable with a minimum length of 8 characters. If there is no pending system name, the value returned is blanks.

CL var for LCLNETID (8) (LCLNETID)

Specifies the name of the CL variable that receives the local network ID. The variable must be a character variable with a minimum length of 8 characters.

CL var for LCLCPNAME (8) (LCLCPNAME)

Specifies the name of the CL variable that receives the local control point name. The variable must be a character variable with a minimum length of 8 characters.

CL var for LCLLOCNAME (8) (LCLLOCNAME)

Specifies the name of the CL variable that receives the default local location name. The variable must be a character variable with a minimum length of 8 characters.

CL var for DFTMODE (8) (DFTMODE)

Specifies the name of the CL variable that receives the default mode name. The variable must be a character variable with a minimum length of 8 characters.

CL var for NODETYPE (8) (NODETYPE)

Specifies the name of the CL variable that receives the APPN node type. The variable must be a character variable with a minimum length of 8 characters.

The following values can be returned in the CL variable:

*ENDNODE
The node does not provide network services to other nodes but can participate in the APPN network by using the services of an attached network server or can operate in a peer environment similar to low entry networking nodes.

*NETNODE
The node provides intermediate routing, route selection services, and distributed directory services for local users and to end nodes and low entry networking nodes that it is serving.

*BEXNODE
The node performs as a branch extender node. The node performs as an end node in the backbone APPN network, and performs as a network node server to end nodes within its local domain.

CL var for DTACPR (10 0) (DTACPR)

Specifies the name of the CL variable that receives the current level of data compression. Specify the name of the decimal variable with a minimum length of 10 digits without decimal positions.

The values that can be returned in the variable as the data compression level are:

0   *NONE - Data compression is not allowed on the session.

-1   *REQUEST - Data compression is requested on the session by the local system. However, the request can be refused or changed to a lower compression level by the remote system. Data compression is allowed on the session if requested by the remote system.

-2   *ALLOW - Data compression is allowed on the session by the local system if requested by a remote system. The local system does not request compression.

-3   *REQUIRE - Data compression is required on the session. If the remote system does not change the levels of compression to the local system's exact requested levels, the session is not established. The data compression levels that the local system requires are the specified levels.

<div align="right">Top</div>

CL var for DTACPRINM (10 0) (DTACPRINM)

Specifies the name of the CL variable that receives the current level of intermediate node data compression. Specify the name of the decimal variable with a minimum length of 10 digits without decimal positions.

The values that can be returned in the variable as the intermediate node data compression levels are:

0   *NONE - The remote systems are not notified of a need to compress data when the system is an SNA intermediate node.

-1   *REQUEST - The remote systems are requested to compress data when the system is an SNA intermediate node.

<div align="right">Top</div>

CL var for MAXINTSSN (5 0) (MAXINTSSN)

Specifies the name of the CL variable that receives the maximum number of intermediate sessions. Specify the name of a decimal variable with a minimum length of 5 digits and no decimal positions.

<div align="right">Top</div>

CL var for RAR (5 0) (RAR)

Specifies the name of the CL variable that receives the route addition resistance. Specify the name of a decimal variable with a minimum length of 5 digits and no decimal positions.

<div align="right">Top</div>

CL var for NETSERVER (85) (NETSERVER)

Specifies the name of the CL variable that receives the list of network node servers. Specify the name of a character variable with a minimum length of 85 characters. If the server name or network ID has fewer characters than the variable allows, the value is padded on the right with blanks. The list contains five node servers. Each server has the form: Network ID (9 characters) followed by the server name (8 characters). There are no separators. The network ID can contain the value *LCLNETID, which specifies that the current network ID is used. If there are less than five node servers specified, the remaining ones contain blanks for a name. As soon as the first blank name is encountered in the list, it is safe to assume that the remaining names are also blank.

<div align="right">Top</div>

CL var for ALRSTS (10) (ALRSTS)

Specifies the name of the CL variable that receives the alert status. Specify the name of a character variable with a minimum length of 10 characters. (If the alert status value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the CL variable:

*ON   Alerts are created by the system for all alert conditions, except unattended conditions.

*OFF   Alerts are not created by the system.

*UNATTEND

Alerts are created by the system for all alert conditions, including messages for which *UNATTEND is specified for the **Alert options (ALROPT)** parameter of the Add Message Description (ADDMSGD) or Change Message Description (CHGMSGD) command.

CL var for ALRPRIFP (10) (ALRPRIFP)

Specifies the name of the CL variable that receives the alert primary focal point. Specify the name of the CL variable with a minimum length of 10 characters. (If the alert primary focal point value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the variable:

*NO   The system is not an alert primary focal point.

*YES   The system is an alert primary focal point.

CL var for ALRDFTFP (10) (ALRDFTFP)

Specifies the name of the CL variable that receives the value for the alert default focal point. Specify a CL variable with a minimum length of 10 characters. (If the alert default focal point value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the variable:

*NO   The system is not an alert default focal point.

*YES   The system is an alert default focal point.

CL var for ALRLOGSTS (7) (ALRLOGSTS)

Specifies the name of the CL variable that receives the alert logging status. The variable must be a character variable with a minimum length of 7 characters.

The following values can be returned in the CL variable:

*NONE

No alerts are logged.

*LOCAL

Only locally created alerts are logged.

*RCV   Only alerts received from other nodes are logged.

*ALL   Both locally created alerts and alerts received from other nodes are logged.

CL var for ALRBCKFP (16) (ALRBCKFP)

Specifies the name of the CL variable that receives the name of the system that provides alert focal point services if the primary focal point is unavailable. Specify the name of a character variable with a minimum length of 16 characters. (If the back up system name has fewer characters than the variable allows, the value is padded on the right with blanks.)

CL var for ALRRQSFP (16) (ALRRQSFP)

Specifies the name of the CL variable that receives the name of the system that is requested to provide alert focal point services. Specify the name of a character variable with a minimum length of 16 characters. (If the requesting system name has fewer characters than the variable allows, the value is padded on the right with blanks.)

CL var for ALRCTLD (10) (ALRCTLD)

Specifies the name of the CL variable that receives the name of the controller through which alert messages are sent to another system when alert processing is active. Specify a CL variable with a minimum length of 10 characters. (If the alert controller name has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the variable:

*NONE
>There is no controller for alerts.

name   Specifies the name of the controller being used for alerts in an alert controller session. This controller is ignored if the system has a primary or default alert focal point (if, for example, the node is in another system's sphere of control).

CL var for ALRHLDCNT (5 0) (ALRHLDCNT)

Specifies the name of the CL variable that receives the maximum number of alerts that are created before the alerts are sent over the alert controller session (ALRCTLD network attribute). The alerts are held (queued) by the system until the specified number of alerts have been created. This parameter can be used to manage alerts that are sent over a limited resource by reducing the number of times alerts are sent.

Note: The ALRHLDCNT network attribute only applies when the ALRCTLD network attribute is used. When management services sessions, APPN, and sphere of control support are used, the ALRHLDCNT value is ignored.

The maximum number of alerts that can be created before the alerts are sent is 32,767. Specify the name of a decimal variable with a total length of 5 digits without decimal positions.

The following values can be returned in the variable:

-2 This value represents *NOMAX. The alerts are held indefinitely. The current alert hold count is the maximum value. The alerts can be sent at a later time by changing the ALRHLDCNT value to a lower value.

0-32767
> Specifies the maximum number of alerts that can be created before being sent. Alerts have a "held" status until the maximum is reached. If the value 0 is specified, alerts are sent as soon as they are created.

CL var for ALRFTR (10) (ALRFTR)

Specifies the name of the CL variable that receives the name of the active alert filter. Specify the name of a character variable with a minimum length of 10 characters. (If the alert filter name has fewer characters than the variable allows, the value is padded on the right with blanks.)

CL var for ALRFTRLIB (10) (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.

CL var for MSGQ (10) (MSGQ)

Specifies the name of the CL variable that receives the system default network message queue name. Specify the name of a character variable with a minimum length of 10 characters. (If the message queue name has fewer characters than the variable allows, the value is padded on the right with blanks.)

CL var for MSGQLIB (10) (MSGQLIB)

Specifies the name of the CL variable that receives the name of the library that contains the system-default network message queue. Specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.)

CL var for OUTQ (10) (OUTQ)

Specifies the name of the CL variable that receives the system default network output queue name. Specify the name of a character variable with a minimum length of 10 characters. (If the output queue name has fewer characters than the variable allows, the value is padded on the right with blanks.)

CL var for OUTQLIB (10) (OUTQLIB)

Specifies the name of the CL variable that receives the name of the library that contains the system-default network output queue. Specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.)

CL var for JOBACN (10) (JOBACN)

Specifies the name of the CL variable that receives the current job action for job streams received through the network. Specify the variable with a minimum length of 10 characters. (If the job action value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the CL variable:

*REJECT
> The input stream is rejected by the system. This action allows you to secure your system from job streams received through the network.

*FILE The input stream is filed in the queue of network files received by the user to whom it was sent. That user can then view, end, or receive the input stream, or submit the input stream to a job queue.

*SEARCH
> The table of network job entries is searched to determine the action taken for the input stream.

CL var for MAXHOP (5 0) (MAXHOP)

Specifies the name of the CL variable that receives the maximum number of times in the SNADS network that a distribution queue originating at this node can be received and rerouted on the path to its final destination. Specify the name of a decimal variable with a total length of 5 digits, and no decimal positions.

CL var for DDMACC (10) (DDMACC)

Specifies the name of the CL variable that receives the current system action for DDM or DRDA requests from other systems. Specify the CL variable with a minimum length of 10 characters. (If the DDM access value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the CL variable:

*REJECT
> This system does not allow DDM and DRDA requests from remote systems. However, this system can still use DDM or DRDA to access files or SQL tables on remote systems. Source (client) systems cannnot access files or SQL tables on any other system that specifies *REJECT.

*OBJAUT
> If the user profile associated with the DDM or DRDA job is authorized to the files, all file or remote SQL requests are accepted. Object authorities such as read, write, or update must also exist for the files.

name   Specifies the name of the customer validation program that can supplement object level security. This user-exit program can restrict user access to *PUBLIC and private files. The target DDM support calls the user program for each reference to a file. The user-exit program indicates to DDM if the request should proceed or end. In the case of DRDA use, the validation program is called only for connection requests, not for individual file access.

CL var for DDMACCLIB (10) (DDMACCLIB)

Specifies the name of the CL variable that receives the name of the library that contains the DDM access program. Specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.) If *REJECT or *OBJAUT is returned for the **CL var for DDMACC (10) (DDMACC)** parameter, the value for this parameter is all blanks.

CL var for PCSACC (10) (PCSACC)

Specifies the name of the CL variable that receives the current system action for iSeries Access requests. Specify a CL variable with a minimum length of 10 characters. (If the iSeries Access access value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the CL variable:

*REJECT
> The system does not allow any iSeries Access requests.

*OBJAUT
> All iSeries Access requests are allowed and controlled by the object authorizations on the system.

*REGFAC
> The registration facility is used to determine exit programs for the different servers. If no program is defined in the registration facility, *OBJAUT is used.

name   The name of the customer supplied iSeries Access host system application exit program that can supplement system object level security.

CL var for PCSACCLIB (10) (PCSACCLIB)

Specifies the name of the CL variable that receives the name of the library that contains the iSeries Access access program. You must specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.) If *REJECT, *REGFAC, or *OBJAUT is returned for the **CL var for PCSACC (10) (PCSACC)** parameter, the value for this parameter will be all blanks.

CL var for DFTNETTYPE (10) (DFTNETTYPE)

Specifies the name of the CL variable that receives the system default value for the Integrated Services Digital Network (ISDN) network type. The operating system no longer uses this network attribute. Changes made to this network attribute have no effect. The variable must be a character variable with a minimum length of 10 characters.

CL var for DFTCNNLST (10) (DFTCNNLST)

Specifies the name of the CL variable that receives the system default value for the ISDN connection list. The operating system no longer uses this network attribute. Changes made to this network attribute have no effect. The variable must be a character variable with a minimum length of 10 characters.

CL var for ALWANYNET (10) (ALWANYNET)

Specifies the name of the CL variable that receives the network attribute that allows the Communications APIs to use other communication transports that are not native for that API. Examples include ICF over TCP/IP or Sockets over SNA. The variable must be a character variable with a minimum length of 10 characters.

CL var for NWSDOMAIN (8) (NWSDOMAIN)

Specifies the name of the CL variable that receives the LAN Server domain to which all integrated PC Servers, also knows as File Server Input/Output Processors(FSIOP), on the system belong. The variable must be a character variable with a minimum length of 8 characters.

CL var for ALWVRTAPPN (10) (ALWVRTAPPN)

Specifies the name of the CL variable that receives the current setting for the virtual APPN support. The character variable must have a minimum length of 10 characters.

CL var for ALWHPRTWR (10) (ALWHPRTWR)

Specifies the name of the CL variable that receives the current setting for the HPR tower transport support. The character variable must have a minimum length of 10 characters.

CL var for VRTAUTODEV (5 0) (VRTAUTODEV)

Specifies the name of the CL variable that receives the current setting for the maximum amount of automatically created APPC devices allowed on a virtual controller. Specify the name of a decimal variable with a total length of 5 digits, and no decimal positions.

CL var for HPRPTHTMR (40) (HPRPTHTMR)

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.

CL var for ALWADDCLU (10) (ALWADDCLU)

Specifies the name of the CL variable that receives the value that specifies whether this system will allow another system to add it as a node in a cluster. The character variable must have a minimum length of 10 characters.

The values that can be returned in the variable for allow add to cluster are:

*NONE
No other system can add this system as a node in a cluster.

*ANY Any other system can add this system as a node in a cluster.

*RQSAUT
Any other system can add this system as a node in a cluster only after the cluster add request has been authenticated.

CL var for MDMCNTRYID (2) (MDMCNTRYID)

Specifies the name of the CL variable that receives the network attribute for the country or region identifier associated with a modem. The character variable must have a minimum length of 2 characters.

MDMCNTRYID defines the country-specific or region-specific default characteristics for modems which are internal to system I/O adapters. This value must be configured correctly to insure proper operation and, in some countries or regions, meet legal requirements. The adapter will fail the vary on of the line if modem country or region ID is not set.

Examples

Example 1: Retrieving Current System Name

```
DCL    VAR(&SNAME)  TYPE(*CHAR)  LEN(8)
RTVNETA    SYSNAME(&SNAME)
```

This command retrieves the current system name.

Example 2: Retrieving Virtual APPN Support, APPC Device Limits, and HPR Path Switch Timers

```
DCL    VAR(&ALWVRTAPPN)  TYPE(*CHAR)  LEN(10)
DCL    VAR(&VRTAUTODEV)  TYPE(*DEC)  LEN(5 0)
DCL    VAR(&HPRPTHTMR)  TYPE(*CHAR)  LEN(40)
RTVNETA    ALWVRTAPPN(&ALWVRTAPPN)  VRTAUTODEV(&VRTAUTODEV)
        HPRPTHTMR(&HPRPTHTMR)
```

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

*ESCAPE Messages

CPF1844
      Cannot access network attribute &1.

Retrieve Object Description (RTVOBJD)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Yes

The Retrieve Object Description (RTVOBJD) command retrieves the description of a specific object to a CL program or REXX procedure.

For parameters that are returned into CL variables by this command, the parameter descriptive title/prompt text lists the minimum length for the CL variable. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

1. You must have execute (*EXECUTE) authority for the library.
2. You must have some authority other than *EXCLUDE authority for the object. If the object is a file, you must have object operational (*OBJOPR) authority for the object.
3. You must have either all object (*ALLOBJ) or audit (*AUDIT) special authority to retrieve a value other than *NOTAVL for the OBJAUD parameter.

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	*Qualified object name*	Required, Positional 1
	Qualifier 1: Object	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OBJTYPE	Object type	*ALRTBL, *AUTL, *BNDDIR, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CNNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DOC, *DTAARA, *DTADCT, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FLR, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *IPXD, *JOBD, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *M36, *M36CFG, *NODGRP, *NODL, *NTBD, *NWID, *NWSCFG, *NWSD, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMQRY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *TIMZON, *USRIDX, *USRPRF, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 2
ASPDEV	ASP device	Single values: *****, *ALLAVL, *CURASPGRP, *SYSBAS Other values: *Element list*	Optional
	Element 1: Device	*Name*	
	Element 2: Search type	***ASP**, *ASPGRP	
RTNLIB	CL var for RTNLIB (10)	*Character value*	Optional
OBJATR	CL var for OBJATR (10)	*Character value*	Optional
USRDFNATR	CL var for USRDFNATR (10)	*Character value*	Optional

Keyword	Description	Choices	Notes
TEXT	CL var for TEXT (50)	*Character value*	Optional
OWNER	CL var for OWNER (10)	*Character value*	Optional
PGP	CL var for PGP (10)	*Character value*	Optional
ASP	CL var for ASP (2 0)	*Decimal number*	Optional
LIBASP	CL var for LIBASP (5 0)	*Decimal number*	Optional
OBJASPDEV	CL var for OBJASPDEV (10)	*Character value*	Optional
OBJASPGRP	CL var for OBJASPGRP (10)	*Character value*	Optional
LIBASPDEV	CL var for LIBASPDEV (10)	*Character value*	Optional
LIBASPGRP	CL var for LIBASPGRP (10)	*Character value*	Optional
OVFASP	CL var for OVFASP (1)	*Character value*	Optional
CRTDATE	CL var for CRTDATE (13)	*Character value*	Optional
CHGDATE	CL var for CHGDATE (13)	*Character value*	Optional
SAVDATE	CL var for SAVDATE (13)	*Character value*	Optional
SAVACTDATE	CL var for SAVACTDATE (13)	*Character value*	Optional
RSTDATE	CL var for RSTDATE (13)	*Character value*	Optional
CRTUSER	CL var for CRTUSER (10)	*Character value*	Optional
CRTSYSTEM	CL var for CRTSYSTEM (8)	*Character value*	Optional
OBJDMN	CL var for OBJDMN (2)	*Character value*	Optional
USEUPD	CL var for USEUPD (1)	*Character value*	Optional
USEDATE	CL var for USEDATE (7)	*Character value*	Optional
USECOUNT	CL var for USECOUNT (5 0)	*Decimal number*	Optional
RESETDATE	CL var for RESETDATE (7)	*Character value*	Optional
STG	CL var for STG (10)	*Character value*	Optional
CPR	CL var for CPR (1)	*Character value*	Optional
SIZE	CL var for SIZE (15 0)	*Decimal number*	Optional
SPCSIZE	CL var for SPCSIZE (15 0)	*Decimal number*	Optional
SPCALIGN	CL var for SPCALIGN (1)	*Character value*	Optional
SAVSIZE	CL var for SAVSIZE (15 0)	*Decimal number*	Optional
SAVCMD	CL var for SAVCMD (10)	*Character value*	Optional
SAVSEQNBR	CL var for SAVSEQNBR (4 0)	*Decimal number*	Optional
SAVLRGSEQ	CL var for SAVLRGSEQ (10 0)	*Decimal number*	Optional
SAVVOL	CL var for SAVVOL (71)	*Character value*	Optional
SAVDEV	CL var for SAVDEV (10)	*Character value*	Optional
SAVF	CL var for SAVF (10)	*Character value*	Optional
SAVFLIB	CL var for SAVFLIB (10)	*Character value*	Optional
SAVLABEL	CL var for SAVLABEL (17)	*Character value*	Optional
SRCF	CL var for SRCF (10)	*Character value*	Optional
SRCFLIB	CL var for SRCFLIB (10)	*Character value*	Optional
SRCMBR	CL var for SRCMBR (10)	*Character value*	Optional
SRCDATE	CL var for SRCDATE (13)	*Character value*	Optional
SYSLVL	CL var for SYSLVL (9)	*Character value*	Optional
COMPILER	CL var for COMPILER (16)	*Character value*	Optional
OBJLVL	CL var for OBJLVL (8)	*Character value*	Optional
ALWAPICHG	CL var for ALWAPICHG (1)	*Character value*	Optional

Keyword	Description	Choices	Notes
APICHG	CL var for APICHG (1)	*Character value*	Optional
USRCHG	CL var for USRCHG (1)	*Character value*	Optional
LICPGM	CL var for LICPGM (16)	*Character value*	Optional
PTF	CL var for PTF (10)	*Character value*	Optional
APAR	CL var for APAR (10)	*Character value*	Optional
OBJAUD	CL var for OBJAUD (10)	*Character value*	Optional
OBJSIG	CL var for OBJSIG (1)	*Character value*	Optional
SYSSIG	CL var for SYSSIG (1)	*Character value*	Optional
MLTSIG	CL var for MLTSIG (1)	*Character value*	Optional
JRNSTS	CL var for JRNSTS (1)	*Character value*	Optional
JRN	CL var for JRN (10)	*Character value*	Optional
JRNLIB	CL var for JRNLIB (10)	*Character value*	Optional
JRNIMG	CL var for JRNIMG (1)	*Character value*	Optional
JRNOMTE	CL var for JRNOMTE (1)	*Character value*	Optional
JRNSTRDATE	CL var for JRNSTRDATE (13)	*Character value*	Optional
STRJRNRCV	CL var for STRJRNRCV (10)	*Character value*	Optional
JRNRCVLIB	CL var for JRNRCVLIB (10)	*Character value*	Optional
RCVLIBASP	CL var for RCVLIBASP (10)	*Character value*	Optional
RCVLIBGRP	CL var for RCVLIBGRP (10)	*Character value*	Optional

Object (OBJ)

Specifies the object for which you want to retrieve information.

This is a required parameter.

Qualifier 1: Object

name　Specify the name of the object for which the description is to be retrieved.

Qualifier 2: Library

*LIBL　All libraries in the library list for the current thread are searched until the first match is found. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*CURLIB
　　The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

name　Specify the name of the library to be searched.

Object type (OBJTYPE)

Specifies the type of the object for which you want to retrieve the information.

This is a required parameter.

object-type
> Specify the type of the object for which information is to be retrieved.

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device name where storage is allocated for the library containing the object. If the library resides in an ASP that is not part of the thread's library name space, this parameter must be specified to ensure the correct library is searched. If this parameter is used when the library qualifier specified for the **Object (OBJ)** parameter is *CURLIB or *LIBL, ASPDEV(*) is the only valid value. This parameter can be specified as a single value or a list of one or two elements.

Single values

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

*ALLAVL
> All available ASPs will be searched. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and all available primary and secondary ASPs (ASPs 33-255 with a status of 'Available').

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

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

Element 1: Device

name

Specify the name of the primary or secondary ASP device to be searched. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Available'. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched.

Note: To specify a specific auxiliary storage pool (ASP) device name when the search type specified for element 2 is *ASP, you must have execute (*EXECUTE) authority for the specific ASP device.

To specify a specific auxiliary storage pool (ASP) device name when the search type specified for element 2 is *ASPGRP, you must have execute (*EXECUTE) authority for each ASP device in the ASP group.

Element 2: Search type

Specifies whether the single ASP device or the entire ASP group named in element 1 is to be searched.

*ASP   Only the single auxiliary storage pool (ASP) device named in element 1 is to be searched.

*ASPGRP

> The entire group of the primary auxiliary storage pool (ASP) device named in element 1 is to be searched.

CL var for RTNLIB (10) (RTNLIB)

Specifies a 10-character variable used to return the name of the library that contains the object. If *LIBL or *CURLIB is specified for the library (qualifier 2) of the **Object (OBJ)** parameter, the value returned is the name of the library where the object was found. If a library name is specified, that name is returned by this parameter.

CL var for OBJATR (10) (OBJATR)

Specifies a 10-character variable used to return an extended attribute of the object such as a program or file type. For example, the variable may be returned with PROD or CLP. No * will precede the value.

CL var for USRDFNATR (10) (USRDFNATR)

Specifies 10-character variable used to return the user-defined attribute of the object. Blanks are returned if the retrieved object does not have a user-defined attribute.

CL var for TEXT (50) (TEXT)

Specifies a 50-character CL variable used to return the text description of the object.

CL var for OWNER (10) (OWNER)

Specifies a 10-character variable used to return the name of the owner of the object.

CL var for PGP (10) (PGP)

Specifies a 10-character variable used to return the name of the user who is the primary group for the object. If there is no primary group for the object, this field contains a value of *NONE.

CL var for ASP (2 0) (ASP)

Specifies a decimal (2 0) variable used to return the number of the auxiliary storage pool (ASP) number for the object. This variable will contain an ASP number up to 32. If an ASP number is larger than 32, -1 is returned in this variable. The OBJASPDEV parameter should be used to return the ASP device name when the ASP number is larger than 32. The following values can be returned:

1 The object is in the system auxiliary storage pool.

2-32 The object is in a basic user auxiliary storage pool.

-1 The object is in a primary or secondary auxiliary storage pool with an ASP number larger than 32. The ASP device name for the primary or secondary ASP can be retrieved with the OBJASPDEV parameter.

CL var for LIBASP (5 0) (LIBASP)

Specifies a decimal (5 0) variable used to return the number of the auxiliary storage pool (ASP) where storage is allocated for the library that contains the object. This variable will contain an ASP number up to 32. If an ASP number is larger than 32, -1 is returned in this variable. The LIBASPDEV parameter should be used to return the ASP device name when the ASP number is larger than 32. The following values can be returned:

1 The library is in the system auxiliary storage pool.

2-32 The library is in a basic user auxiliary storage pool.

-1 The library is in a primary or secondary auxiliary storage pool with an ASP number larger than 32. The ASP device name for the primary or secondary ASP can be retrieved with the LIBASPDEV parameter.

CL var for OBJASPDEV (10) (OBJASPDEV)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the object. The following special values can be returned:

*N The name of the ASP device cannot be determined.

*SYSBAS
  The object is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

CL var for OBJASPGRP (10) (OBJASPGRP)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) group where storage is allocated for the object. The ASP group name is the name of the primary ASP within the ASP group. The value returned may be the same as the value returned for the OBJASPDEV parameter. The following special values can be returned:

*N The name of the primary ASP device cannot be determined.

*SYSBAS
  The object is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

CL var for LIBASPDEV (10) (LIBASPDEV)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the library that contains the object. The following special values can be returned:

*N   The name of the ASP device cannot be determined.

*SYSBAS

The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

CL var for LIBASPGRP (10) (LIBASPGRP)

Specifies a 10-character variable used to return the name of the primary auxiliary storage pool (ASP) group where storage is allocated for the library that contains the object. The ASP group name is the name of the primary ASP within the ASP group. The value returned may be the same as the value returned for the LIBASPDEV parameter. The following special values can be returned:

*N   The name of the primary ASP device cannot be determined.

*SYSBAS

The object is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

CL var for OVFASP (1) (OVFASP)

Specifies a 1-character variable used to return the Object Overflowed auxiliary storage pool (ASP) flag.

'1'   The object overflowed the ASP in which it resides.

'0'   The object does not overflow the ASP. It is not possible for an object residing in the system ASP (ASP 1) or in a primary or secondary ASP (ASPs 33-255) to overflow its ASP, therefore, a value of '0' is always returned for objects in the system ASP (ASP 1) or in a primary or secondary ASP (ASPs 33-255).

CL var for CRTDATE (13) (CRTDATE)

Specifies a 13-character variable used to return the date and time the object was created. A value is returned in the form CYYMMDDHHMMSS where C = century; '0' indicates years 19xx and '1' indicates years 20xx; YY = year, MM = month, DD = day, HH = hour, MM = minutes and SS = seconds.

CL var for CHGDATE (13) (CHGDATE)

Specifies a 13-character variable used to return the date and time the object was last changed. The variable is returned in the same format as the CRTDATE parameter or is returned blank if the object has not been changed.

CL var for SAVDATE (13) (SAVDATE)

Specifies a 13-character variable used to return the date and time the object was last saved. The variable is returned in the same format as the CRTDATE parameter or is returned blank if the object has not been saved.

CL var for SAVACTDATE (13) (SAVACTDATE)

Specifies a 13-character variable used to return the date and time the object was last saved during a save operation with the SAVACT parameter specified as something other than *NO. This date/time indicates when the object itself was saved; the SAVDAT parameter indicates when the save operation was started. The variable is returned in the same format as the CRTDATE parameter or is returned blank if the object has not been saved or if SAVACT(*NO) was specified for the last save operation for the object.

CL var for RSTDATE (13) (RSTDATE)

Specifies a 13-character variable used to return the date and time the object was last restored. The variable is returned in the same format as the CRTDATE parameter or is returned blank if the object has not been restored.

CL var for CRTUSER (10) (CRTUSER)

Specifies a 10-character variable used to return the name of the user that created the object.

CL var for CRTSYSTEM (8) (CRTSYSTEM)

Specifies an 8-character variable used to return the name of the system on which the object was created.

CL var for OBJDMN (2) (OBJDMN)

Specifies a 2-character variable used to return the object domain value for the object. The following values can be returned:

*U   The object is a user domain object.

*S   The object is a system domain object.

CL var for USEUPD (1) (USEUPD)

Specifies a 1-character variable used to return whether the object usage information is updated for this object type. The following values can be returned:

Y   Object usage information is updated for this object type.

N   Object usage information is not updated for this object type. If 'N' is returned, the last used date for the object is blank and the number of days the object has been used is zero (0).

CL var for USEDATE (7) (USEDATE)

Specifies a 7-character variable used to return the date the object was last used. The date is returned in the form CYYMMDD or is returned blank if the object does not have a last used date.

CL var for USECOUNT (5 0) (USECOUNT)

Specifies a decimal (5 0) variable used to return the number of days the object has been used. If the object does not have a last used date, zero (0) is returned.

CL var for RESETDATE (7) (RESETDATE)

Specifies a 7-character variable used to return the date the days used count was last reset to zero (0). The date is returned in the form CYYMMDD or is returned blank if the days used count has not been reset.

CL var for STG (10) (STG)

Specifies a 10-character variable used to return the storage status of the object data. The following values can be returned:

*FREE The object data has been freed and the object is suspended.

*KEEP
        The object data has not been freed and the object has not been suspended.

CL var for CPR (1) (CPR)

Specifies a 1-character variable used to return the compression status of the object. The following values can be returned:

Y   The object is compressed.

X   The object is ineligible for compression.

N   The object is permanently decompressed.

T   The object is temporarily decompressed.

F   The object is eligible for compression but is saved with storage freed.

CL var for SIZE (15 0) (SIZE)

Specifies a decimal (15 0) variable used to return the size of the object in bytes. This value includes the value returned by the SPCSIZE parameter.

CL var for SPCSIZE (15 0) (SPCSIZE)

Specifies a decimal (15 0) variable used to return the size of the primary associated space of object in bytes. If the object has no associated space, zero (0) is returned.

CL var for SPCALIGN (1) (SPCALIGN)

Specifies a 1-character variable used to return whether the space associated with the object has been optimally aligned. Optimum alignment may allow for better performance of applications that use the object. The following values can be returned:

'0'   The space associated with the object has not been optimally aligned.

'1'   The space associated with the object has been optimally aligned.

'2'   There is not a space associated with the object.

CL var for SAVSIZE (15 0) (SAVSIZE)

Specifies a decimal (15 0) variable used to return the size of the object in bytes at the time of the last save operation. If the object has not been saved, zero (0) is returned.

CL var for SAVCMD (10) (SAVCMD)

Specifies a 10-character variable used to return the command used to save the object. If the object has not been saved, the variable is returned blank.

CL var for SAVSEQNBR (4 0) (SAVSEQNBR)

Specifies a decimal (4 0) variable used to return the tape sequence number assigned when the object was saved on tape. This variable will contain a sequence number up to 9999. If the object has not been saved or was not saved to tape, zero (0) is returned. If a sequence number is actually greater than 9999, -5 is returned in this variable. The SAVLRGSEQ parameter should be used to return a sequence number which can be larger than 9999.

CL var for SAVLRGSEQ (10 0) (SAVLRGSEQ)

Specifies a decimal (10 0) variable used to return the tape sequence number (similar to the SAVSEQNBR parameter). This variable can contain a larger tape sequence number than the SAVSEQNBR parameter.

CL var for SAVVOL (71) (SAVVOL)

Specifies a 71-character variable used to return the tape, diskette, or optical volumes used for saving the object. The variable returns a maximum of 10 six-character volumes. The volume IDs begin in character positions 1, 8, 15, 22, 29, 36, 43, 50, 57, and 64. Each volume ID entry is separated by a single character. If the object was saved in parallel format, the separator character contains a '2' before the first volume in the second media file, a '3' before the third media file, and so on, up to a '0' before the tenth media file. Otherwise, the separator characters are blank. If more than 10 volumes are used and the object was saved in serial format, '1' is returned in the 71st character of the variable. If the object was saved in parallel format, a '2' is returned in the 71st character of the variable. Otherwise, the 71st character is blank. If the object was last saved to a save file or was never saved, the variable is returned blank.

CL var for SAVDEV (10) (SAVDEV)

Specifies a 10-character variable used to return the type of the device to which the object was last saved. The variable is returned with one of the following values, dependent on the device used for the last save operation:

- *SAVF for a save file
- *DKT for a diskette
- *TAP for a tape
- *OPT for an optical volume
- The variable is returned blank if the object was not saved.

CL var for SAVF (10) (SAVF)

Specifies a 10-character variable used to return the name of the save file if the object was saved to a save file. If the object was not saved to a save file, the variable is returned blank.

CL var for SAVFLIB (10) (SAVFLIB)

Specifies a 10-character variable used to return the name of the library that contains the save file if the object was saved to a save file. If the object is not saved to a save file, the variable is returned blank.

CL var for SAVLABEL (17) (SAVLABEL)

Specifies a 17-character variable used to return the file label used when the object was saved. If the object is not saved to tape, to diskette, or to an optical volume, the variable is returned blank. The value returned corresponds to the value specified for the LABEL parameter on the command used to save the object.

CL var for SRCF (10) (SRCF)

Specifies a 10-character variable used to return the name of the source file that was used to create the object. If no source file was used to create the object, the variable is returned blank. For ILE *PGM and *SRVPGM objects, the source file and member are blank. The source file information is stored with the *MODULE object.

CL var for SRCFLIB (10) (SRCFLIB)

Specifies a 10-character variable used to return the name of the library that contains the source file that was used to create the object. If no source file is used to create the object, the variable is returned blank.

CL var for SRCMBR (10) (SRCMBR)

Specifies a 10-character variable used to return the name of the member in the source file (SRCF parameter). If no source file is used to create the object, the variable is returned blank.

CL var for SRCDATE (13) (SRCDATE)

Specifies a 13-character variable used to return the date and time the member in the source file was last updated. The variable is returned in the same format as the CRTDATE parameter or is returned blank if the member is not updated.

CL var for SYSLVL (9) (SYSLVL)

Specifies a 9-character variable used to return the level of the operating system when the object was created. The variable is returned with a 3-character version level starting in character position 1, a 3-character release level starting in character position 4, and a 3-character modification level starting in character position 7. The first character of the version level is always the letter 'V'; the first character of the release level is always the letter 'R'; the first character of the modification level is always the letter 'M'.

CL var for COMPILER (16) (COMPILER)

Specifies a 16-character variable used to return the licensed program identifier, version level, release level, and modification level of the compiler. The variable is returned with the 7-character program identifier starting in character position 1, the 3-character version level in character position 8, the 3-character release level in character position 11, and the 3-character modification level in character position 14. The first character of the version level is always the letter 'V'; the first character of the release level is always the letter 'R'; the first character of the modification level is always the letter 'M'. If no compiler was used, the variable is returned blank.

CL var for OBJLVL (8) (OBJLVL)

Specifies a 8-character variable used to return the object control level for the created object.

CL var for ALWAPICHG (1) (ALWAPICHG)

Specifies a 1-character variable used to return the Allow Change by Program flag. The following values can be returned:

'1'   The object can be changed with the Change Object Description (QLICOBJD) API.

'0'   The object cannot be changed with the API.

CL var for APICHG (1) (APICHG)

Specifies a 1-character variable used to return the Changed by Program flag. The following values can be returned:

'1'   The object has been modified with the Change Object Description (QLICOBJD) API.

'0'   The object has not been changed by the API.

CL var for USRCHG (1) (USRCHG)

Specifies a 1-character variable used to return whether the object has been modified by the user. The following values can be returned:

'1'   The object has been modified by the user.

'0'   The object has not been modified by the user.

CL var for LICPGM (16) (LICPGM)

Specifies a 16-character variable used to return the name, version level, release level, and modification level of the licensed program if the retrieved object is part of a licensed program. The variable is returned with the 7-character name starting in character position 1, the 3-character version level in character position 8, the 3-character release level in character position 11, and the 3-character modification level in character position 14. The first character of the version level is always the letter 'V'; the first character of the release level is always the letter 'R'; the first character of the modification level is always the letter 'M'. If the retrieved object is not part of a licensed program, the variable is returned blank.

CL var for PTF (10) (PTF)

Specifies a 10-character variable used to return the Program Temporary Fix number that resulted in the creation of the retrieved object. For user-created objects, the variable is returned blank.

CL var for APAR (10) (APAR)

Specifies a 10-character variable used to return the Authorized Program Analysis Report identification (APAR ID). The variable is returned with the APAR ID that caused this object to be patched. If the object has not been changed as a result of an APAR, the variable is returned blank.

This field is not updated when IBM-supplied Program Temporary Fixes are applied. The field is changed in the following situations:

- The default for a command is changed with the Change Command Default (CHGCMDDFT) CL command. The field is set to CHGDFT.
- The Change Object Description (QLICOBJD) API can change this field to any value.

CL var for OBJAUD (10) (OBJAUD)

Specifies a 10-character variable used to return the auditing value of the object. A value of *NOTAVL will be returned if you do not have either all object (*ALLOBJ) or audit (*AUDIT) special authority. The values that can be returned include *NONE, *USRPRF, *CHANGE, *ALL, and *NOTAVL. See the **Object auditing value (OBJAUD)** parameter on the Change Object Audit (CHGOBJAUD) command for more information.

CL var for OBJSIG (1) (OBJSIG)

Specifies a 1-character variable used to return whether the object has a digital signature. The following values can be returned:

'1'   The object has a digital signature.

'0'   The object does not have a digital signature.

CL var for SYSSIG (1) (SYSSIG)

Specifies a 1-character variable used to return whether the object is signed by a source that is trusted by the system. The following values can be returned:

'1'   The object is signed by a source that is trusted by the system. If the object has multiple signatures, at least one of the signatures came from a source that is trusted by the system.

'0'   None of the object signatures came from a source that is trusted by the system.

CL var for MLTSIG (1) (MLTSIG)

Specifies a 1-character variable used to return whether the object has more than one digital signature. The following values can be returned:

'1'   The object has more than one digital signature. If the SYSSIG parameter has a value of '1', at least one of the signatures is from a source trusted by the system.

'0'   The object has only one digital signature, or does not have a digital signature. Refer to the OBJSIG parameter to determine whether the object has a digital signature.

CL var for JRNSTS (1) (JRNSTS)

Specifies a 1-character variable used to return the current journaling status of the object. The following values can be returned:

'1'   The object is currently being journaled.

'0'   The object is currently not being journaled.

Note: Other journal fields may contain data even though the object is not currently being journaled.

CL var for JRN (10) (JRN)

Specifies a 10-character variable used to return the name of the current or last journal. If the object has never been journaled, the variable is returned blank.

CL var for JRNLIB (10) (JRNLIB)

Specifies a 10-character variable used to return the name of the library that contains the journal. If the object has never been journaled, the variable is returned blank.

CL var for JRNIMG (1) (JRNIMG)

Specifies a 1-character a variable used to return the journal image information. The following values can be returned dependent upon what images are generated for changes to the object:

'0'   Only *after* images are written to the journal for changes to the object.

'1'   Both *before* and *after* images are written to the journal for changes to the object.

If the object has never been journaled, the variable is returned blank.

CL var for JRNOMTE (1) (JRNOMTE)

Specifies 1-character variable used to return information regarding journal entries to be omitted. The following values can be returned for the journal entries to be omitted:

'1'   *open* and *close* operations on the specified objects do not generate *open* and *close* journal entries.

'0'   No journal entries are omitted.

If the object has never been journaled, the variable is returned blank.

CL var for JRNSTRDATE (13) (JRNSTRDATE)

Specifies a 13-character variable used to return the date and time journaling was last started. The value is returned in the same format as the CRTDATE parameter or is returned blank if the object has never been journaled.

CL var for STRJRNRCV (10) (STRJRNRCV)

Specifies a 10-character variable used to return the name of the oldest journal receiver needed to successfully use the Apply Journaled Changes (APYJRNCHG) or Remove Journaled Changes (RMVJRNCHG) command. This field will be blank if the object has never been journaled.

For a file object, the journal receiver will contain the entry representing the start-of-the-save operation. However, if there are members within the file that contain partial transactions, then those members may require an earlier journal receiver. Use the Display File Description (DSPFD) command to determine the partial transaction state of the members of the file.

CL var for JRNRCVLIB (10) (JRNRCVLIB)

Specifies a 10-character variable used to return the name of the library that contains the starting journal receiver. This field will be blank if the object has never been journaled.

CL var for RCVLIBASP (10) (RCVLIBASP)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the library that contains the starting journal receiver. This field will be blank if the object has never been journaled. The following special values can be returned:

*N   The name of the ASP device cannot be determined.

*SYSBAS

> The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

CL var for RCVLIBGRP (10) (RCVLIBGRP)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) group where storage is allocated for the library that contains the starting journal receiver. The ASP group name is the name of the primary ASP within the ASP group. The value returned may be the same as the value returned for the RCVLIBASP parameter. This field will be blank if the object has never been journaled. The following special values can be returned:

*N   The name of the ASP device cannot be determined.

*SYSBAS

> The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

Examples

Jane Brown enters the following command to create a library:

```
CRTLIB   LIB(PGMLIB)  TYPE(*PROD)
         TEXT('Library for test programs')
```

Later, Jane, or anyone else with the proper authority, can retrieve the attributes of the library in a CL program as shown below:

```
DCL    &CRTDATE   *CHAR  13
DCL    &OWN       *CHAR  10
DCL    &RTNLIB    *CHAR  10
DCL    &OBJASPDEV *CHAR  10
 :
RTVOBJD   OBJ(*LIBL/PGMLIB) OBJTYPE(*LIB) TEXT(&TEXT)  +
          CRTDATE(&CRTDATE) OWNER(&OWN)                +
          RTNLIB(&RTNLIB) OBJASPDEV(&OBJASPDEV)
```

The values returned in the variables of the CL program are shown below:

```
&TEXT      =  Library for test programs
&CRTDATE   =  0900211130000
&OWN       =  JBROWN
&RTNLIB    =  QSYS
&OBJASPDEV =  *SYSBAS
```

The value returned in the variable &CRTDATE indicates that PGMLIB was created on the eleventh day of February, 1990, at 1300 hours. The value returned in the variable &OWN indicates that the library was created by user profile JBROWN. The value returned in the variable &RTNLIB indicates that PGMLIB is in library QSYS. The value returned in the variable &OBJASPDEV indicates that storage for PGMLIB is allocated from *SYSBAS which includes the system auxiliary storage pool (ASP 1) and any defined basic user ASPs (ASPs 2-32).

Error messages

CPFB8ED

Device description &1 not correct for operation.

CPF2115

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

CPF2150

Object information function failed.

CPF2151

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

CPF2173

Value for ASPDEV not valid with special value for library.

CPF218C

&1 not a primary or secondary ASP.

CPF218D

&1 not a primary ASP when *ASPGRP specified.

CPF2451

Message queue &1 is allocated to another job.

CPF3202

File &1 in library &2 in use.

CPF3203

Cannot allocate object for file &1 in &2.

CPF36F7

Message queue QSYSOPR is allocated to another job.

CPF980B

Object &1 in library &2 not available.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9807

One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

CPF9810

Library &1 not found.

CPF9811

Program &1 in library &2 not found.

CPF9812

File &1 in library &2 not found.

CPF9814

Device &1 not found.

CPF9820

>Not authorized to use library &1.

CPF9821

>Not authorized to program &1 in library &2.

CPF9822

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

CPF9825

>Not authorized to device &1.

CPF9830

>Cannot assign library &1.

CPF9831

>Cannot assign device &1.

CPF9833

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

Top

Retrieve PDG Profile (RTVPDGPRF)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Print Descriptor Group Profile (RTVPDGPRF) command is used in a CL program or REXX procedure to retrieve one or more of the print descriptor group profile values associated with a user profile. The values are returned in the specified CL variables for the desired user.

Restrictions:

1. The program must have *OBJOPR authority to the command.
2. The program must have *READ authority to the user's profile.

Parameters

Keyword	Description	Choices	Notes
USER	User profile	*Name*, *CURRENT	Required, Positional 1
RTNUSER	CL var for RTNUSER	*Character value*	Optional
PDG	CL var for PDG	*Character value*	Optional
PDGLIB	CL var for PDGLIB	*Character value*	Optional
PRTD	CL var for PRTD	*Character value*	Optional

User profile (USER)

Specifies the user profile name to be checked for its print descriptor group (PDG) profile.

The possible values are:

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

user-name
    Specify the user profile to check.

CL var for RTNUSER (RTNUSER)

In control language (CL) programs, specifies the name of the 10-character variable used to get the name of the user profile for which information is requested.

CL var for PDG (PDG)

In control language (CL) programs, specifies the name of the 10-character variable used to get the PDG profile of the user for which information is requested.

CL var for PDGLIB (PDGLIB)

In control language (CL) programs, specifies the name of the 10-character variable used to get the library of the PDG profile of the user for which information is requested.

CL var for PRTD (PRTD)

In control language (CL) programs, specifies the name of the 256-character variable used to get the print descriptor name from the PDG profile of the user for which information is requested.

Examples

Assume a user with *OBJMGT authority entered the following command:

```
CHGUSRPRF   USER(JWONG)  PDG(*LIBL/TAXFORMS)  PRTD(FORM_C1)
```

Also assume the program with *OBJMGT authority contains the following commands and declarations:

```
DCL   VAR(&USER)      TYPE(*CHAR)  LEN(10)
DCL   VAR(&GROUP)     TYPE(*CHAR)  LEN(10)
DCL   VAR(&LIBRARY)   TYPE(*CHAR)  LEN(10)
DCL   VAR(&DESCRIPT)  TYPE(*CHAR)  LEN(256)
RTVPDGPRF   USER(JWONG)  RTNUSER(&USER)  PDG(&GROUP)  PDGLIB(&LIBRARY)
            PRTD(&DESCRIPT)
```

When the above program is called, the following values are returned:

```
&USER       'JWONG      '
&GROUP      'TAXFORMS   '
&LIBRARY    'TAXLIB     '
&DESCRIPT   'FORM_C1 ... '
```

Note: The value returned in variable &DESCRIPT is FORM_C1 followed by 249 blanks.

Error messages

*ESCAPE Messages

CPF2204

      User profile &1 not found.

CPF2217

      Not authorized to user profile &1.

CPF2247

      Internal security object not available. Reason code &1.

Retrieve Power Schedule Entry (RTVPWRSCDE)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Power On/Off Schedule Entry (RTVPWRSCDE) command retrieves a power on/off schedule entry value for use in a CL or REXX program. The value is returned (copied) to the specified CL variable in the program.

Restrictions:

1. To use this command, you must have use (*USE) authority to the Display Power On/Off Schedule (DSPPWRSCD) command.

Parameters

Keyword	Description	Choices	Notes
DAY	Day	*Date*, ***TODAY**, *SUN, *MON, *TUE, *WED, *THU, *FRI, *SAT	Optional, Positional 1
PWRONTIME	CL var for PWRONTIME (6)	*Character value*	Optional
PWROFFTIME	CL var for PWROFFTIME (6)	*Character value*	Optional
DAYDESC	CL var for DAYDESC (38)	*Character value*	Optional
MSGITV	CL var for MSGITV (2 0)	*Decimal number*	Optional

Day (DAY)

Specifies the day for which you are retrieving a power on/off schedule entry.

*TODAY
> The current date's schedule entry is retrieved.

*SUN The schedule entry for Sunday is retrieved.

*MON
> The schedule entry for Monday is retrieved.

*TUE The schedule entry for Tuesday is retrieved.

*WED The schedule entry for Wednesday is retrieved.

*THU The schedule entry for Thursday is retrieved.

*FRI The schedule entry for Friday is retrieved.

*SAT The schedule entry for Saturday is retrieved.

date Specify the date for which a schedule entry is to be retrieved. The date must be specified in the same format as specified by your job attributes.

CL var for PWRONTIME (6) (PWRONTIME)

Specifies the name of the CL variable that receives the power on time. The variable named has a minimum length of 6 characters. The special value *NONE or the time in the format **hhmmss**, where **hh = hours,mm = minutes**, and **ss = seconds**, is returned.

CL var for PWROFFTIME (6) (PWROFFTIME)

Specifies the name of the CL variable that receives the power off time. The variable named has a minimum length of 6 characters. The special value *NONE or the time in the format **hhmmss**, where **hh = hours,mm = minutes**, and **ss = seconds**, is returned.

CL var for DAYDESC (38) (DAYDESC)

Specifies the name of the CL variable that receives the day description value. The day description is an explanation of the power on/off schedule for that date. The variable named has a minimum length of 38 characters.

CL var for MSGITV (2 0) (MSGITV)

Specifies the name of the CL variable that receives the message interval value. The message interval is the number of minutes before the scheduled power off that a message is sent to all work stations warning users of the power off. The variable named has a minimum length of 2 characters.

Examples

Example 1: Retrieving Today's Schedule Entry

```
DCL   VAR(&ONTIME)  TYPE(*CHAR) LEN(6)
DCL   VAR(&OFFTIME) TYPE(*CHAR) LEN(6)
RTVPWRSCDE   DAY(*TODAY) PWRONTIME(&ONTIME) +
             PWROFFTIME(&OFFTIME)
```

This command retrieves the power on and off times for today.

Example 2: Retrieving Tuesday's Schedule Entry

```
DCL   VAR(&ONTIME)  TYPE(*CHAR) LEN(6)
DCL   VAR(&OFFTIME) TYPE(*CHAR) LEN(6)
RTVPWRSCDE   DAY(*TUE) PWRONTIME(&ONTIME)  PWROFFTIME(&OFFTIME)
```

This command retrieves the power on and power off times for Tuesday's.

Error messages

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.

Top

Retrieve Query Management Form (RTVQMFORM)

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

The Retrieve Query Management Form (RTVQMFORM) command allows you to retrieve encoded form source records from a query management form (QMFORM) object. The source records are placed into a source file member that can be edited.

Form source can also be retrieved from a query definition (QRYDFN) object when the specified QMFORM does not exist.

Parameters

Keyword	Description	Choices	Notes
QMFORM	Query management report form	*Qualified object name*	Required, Positional 1
	Qualifier 1: Query management report form	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCFILE	Source file	*Qualified object name*	Required, Positional 2
	Qualifier 1: Source file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***QMFORM**	Optional, Positional 3
ALWQRYDFN	Allow information from QRYDFN	***NO**, *YES, *ONLY	Optional

Query management report form (QMFORM)

Specifies the query management form object whose source is being retrieved.

This is a required parameter.

Qualifier 1: Query management report form

name   Specify the name of the form being retrieved.

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Source file (SRCFILE)

Specifies the previously-created source physical file into which the encoded form source records are to be written.

This is a required parameter.

Qualifier 1: Source file

name   Specify the name of the source file that is to receive the form source.

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Source member (SRCMBR)

Specifies the source physical file member into which the encoded form source records are to be written. If a source file member name is not specified, then the form name specified on the QMFORM parameter is used.

If the member existed before running this command, it is cleared before any source statements are written into it. If the member does not exist, it is created.

*QMFORM
>   The member name is the same as the form name specified on the QMFORM parameter.

name   Specify the name of the member to receive the form source.

Allow information from QRYDFN (ALWQRYDFN)

Specifies whether form information is taken from a QRYDFN object when a query management form (QMFORM) object cannot be found using the specified object name. Any information that has to be derived in this way is discarded when the command has completed processing. No query management object is created.

*NO   Information is not taken from a QRYDFN object.

*YES   Information is taken from a QRYDFN object when the specified QMQRY object is not found.

*ONLY
>   Information is taken only from a QRYDFN. Query management objects are ignored.

Examples

Example 1: Retrieving Encoded Form Source

```
RTVQMFORM   QMFORM(RPTLIB/SALFORM)   SRCFILE(FORMS)
            SRCMBR(EMPFORM)
```

This command retrieves the encoded form source from the form named SALFORM located in the RPTLIB library. The encoded form source records that are retrieved are placed into the newly created or cleared member EMPFORM in the first file named FORMS in the user's library list.

Example 2: Retrieving Source From Either the QMFORM or the QRYDFN

```
RTVQMFORM   QMFORM(RPTLIB/SALFORM)   SRCFILE(FORMS)
            SRCMBR(EMPFORM)   ALWQRYDFN(*YES)
```

This command retrieves the encoded form source from the query management form (QMFORM) named SALFORM located in the RPTLIB library. If there is no QMFORM object named SALFORM in the RPTLIB library, then the form source is retrieved from the query definition (QRYDFN) named SALFORM in the RPTLIB library. The encoded form source records that are retrieved are placed into the first file named FORMS in the user's library list.

Error messages

*ESCAPE Messages

QWM2701
> &1 command failed.

QWM2703
> &1 command ended.

QWM2705
> Source file &1 in &2 not available.

Retrieve Query Mgmt Query (RTVQMQRY)

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

The Retrieve Query Management Query (RTVQMQRY) command allows you to retrieve Structured Query Language (SQL) source from a query management query (QMQRY) object. The source records are placed into an editable source file member.

You can also retrieve query source from a query definition (QRYDFN) object when the specified QMQRY does not exist.

Parameters

Keyword	Description	Choices	Notes
QMQRY	Query management query	*Qualified object name*	Required, Positional 1
	Qualifier 1: Query management query	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCFILE	Source file	*Qualified object name*	Required, Positional 2
	Qualifier 1: Source file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***QMQRY**	Optional, Positional 3
ALWQRYDFN	Allow information from QRYDFN	***NO**, *YES, *ONLY	Optional

Query management query (QMQRY)

Specifies the query management query whose source is to be retrieved.

This is a required parameter.

Qualifier 1: Query management query

name   Specify the name of the query whose source is to be retrieved.

Qualifier 2: Library

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

*CURLIB

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

name   Specify the name of the library to be searched.

Source file (SRCFILE)

Specifies the previously-created source physical file into which the query source records are to be written.

This is a required parameter.

Qualifier 1: Source file

name   Specify the name of the source file to receive the query source being retrieved.

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Source member (SRCMBR)

Specifies the source file member into which the query source records are being written. If the member existed before running this command, it is cleared before any source records are written to it. If the member does not exist, it is created.

*QMQRY
>   The member name is the same as the query name specified on the QMQRY parameter.

name   Specify the name of the file member to receive the query source.

Allow information from QRYDFN (ALWQRYDFN)

Specifies whether query information is taken from a query definition (QRYDFN) object when a query management (QMQRY) object cannot be found using the specified object name. Any information that has to be derived in this way is discarded when the command has completed processing. No query management object is created.

*NO   Information is not taken from a QRYDFN object.

*YES   Information is taken from a QRYDFN object when the specified QMQRY object is not found.

*ONLY
>   Information is taken only from a QRYDFN object. Query management objects are ignored.

Examples

Example 1: Retrieving SQL Source

```
RTVQMQRY   QMQRY(RPTLIB/SALQRY)   SRCFILE(QRYS)   SRCMBR(EMPQRY)
```

This command retrieves the source from the query named SALQRY located in the RPTLIB library. The source records that are retrieved are placed into the newly created or cleared member EMPQRY in the first file named QRYS in the user's library.

Example 2: Retrieving Source From Either the QMQRY or the QRYDFN

```
RTVQMQRY   QMQRY(RPTLIB/SALQRY)   SRCFILE(QRYS)   SRCMBR(EMPQRY)
           ALWQRYDFN(*YES)
```

This command retrieves the source from the query management query (QMQRY) named SALQRY in the RPTLIB library. If there is no QMQRY object named SALQRY in the RPTLIB library, then the query source is retrieved from the query definition (QRYDFN) named SALQRY in the RPTLIB library. The source records are placed into the newly created or cleared member EMPQRY in the first file named QRYS in the user's library list.

Error messages

*ESCAPE Messages

QWM2701
&1 command failed.

QWM2703
&1 command ended.

QWM2705
Source file &1 in &2 not available.

Retrieve S/36 Environment Attr (RTVS36A)

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

Threadsafe: No

The Retrieve System/36 Attributes (RTVS36A) command allows the user to retrieve specific attribute information about the System/36 environment configuration and provide that information to a specified variable of a CL program or REXX procedure. Information about one or more attributes can be retrieved.

More information about the System/36 attributes that can be retrieved is in the help information for the Change System/36 Attributes (CHGS36A) command.

Top

Parameters

Keyword	Description	Choices	Notes
ENV	Environment name	*Name*, **#LIBRARY**	Optional, Key
SLIB	CL var for SLIB (8)	*Character value*	Optional
FLIB	CL var for FLIB (10)	*Character value*	Optional
LIBL	CL var for LIBL (4)	*Character value*	Optional
DATDIFF	CL var for DATDIFF (4)	*Character value*	Optional
S36ESHARE	CL var for S36ESHARE (4)	*Character value*	Optional
RCDBLK	CL var for RCDBLK (4)	*Character value*	Optional
CACHEDLTF	CL var for CACHEDLTF (4)	*Character value*	Optional
LPPAGE	CL var for LPPAGE (3)	*Character value*	Optional
FORMTYPE	CL var for FORMTYPE (4)	*Character value*	Optional
DFTMSGACN	CL var for DFTMSGACN (9)	*Character value*	Optional
HALTOPT	CL var for HALTOPT (4)	*Character value*	Optional
EVKJOBINIT	CL var for EVKJOBINIT (6)	*Character value*	Optional
EVKJOBPOL	CL var for EVKJOBPOL (8)	*Character value*	Optional
EVKJOBPTY	CL var for EVKJOBPTY (10)	*Character value*	Optional
SRCRCDLEN	CL var for SRCRCDLEN (3)	*Character value*	Optional
CHGACT	CL var for CHGACT (4)	*Character value*	Optional
ADDS36ONLY	CL var for ADDS36ONLY (4)	*Character value*	Optional
ICFSUBST	CL var for ICFSUBST (4)	*Character value*	Optional
MRTUSRPRF	CL var for MRTUSRPRF (8)	*Character value*	Optional
MRTAUT	CL var for MRTAUT (8)	*Character value*	Optional
MRTDLY	CL var for MRTDLY (5)	*Character value*	Optional
MRTJOBINIT	CL var for MRTJOBINIT (6)	*Character value*	Optional
MRTJOBPOL	CL var for MRTJOBPOL (8)	*Character value*	Optional
MRTJOBPTY	CL var for MRTJOBPTY (10)	*Character value*	Optional

Environment name (ENV)

Specifies the name of the System/36 environment from which you are retrieving attributes. The value is #LIBRARY and cannot be changed.

This is a required parameter.

CL var for SLIB (8) (SLIB)

Specifies the name of an 8-character variable to receive the name of the default session library for jobs running in the System/36 environment.

CL var for FLIB (10) (FLIB)

Specifies the name of a 10-character variable to receive the name of the default files library for jobs running in the System/36 environment.

CL var for LIBL (4) (LIBL)

Specifies the name of a 4-character variable to receive information on whether the library list is used for jobs running in the System/36 environment. A value of *YES or *NO is returned in the variable.

CL var for DATDIFF (4) (DATDIFF)

Specifies the name of a 4-character variable to receive information on whether files with the same name but different dates can be used for jobs running in the System/36 environment. A value of *YES or *NO is returned in the variable.

CL var for S36ESHARE (4) (S36ESHARE)

Specifies the name of a 4-character variable to receive information on whether programs share an open data path (ODP) to database files opened in the System/36 environment. A value of *YES or *NO is returned in the variable.

CL var for RCDBLK (4) (RCDBLK)

Specifies the name of a 4-character variable to receive information on whether record blocking is used for sequential database files sharing an open data path in the System/36 environment. A value of *YES or *NO is returned in the variable.

CL var for CACHEDLTF (4) (CACHEDLTF)

Specifies the name of a 4-character variable to receive information on whether deleted files are stored in a cache in the System/36 environment. A value of *YES or *NO is returned in the variable.

CL var for LPPAGE (3) (LPPAGE)

Specifies the name of a 3-character variable to receive the number of lines printed on a page for jobs running in the System/36 environment. A value ranging from 1 through 112 is returned in the variable.

CL var for FORMTYPE (4) (FORMTYPE)

Specifies the name of a 4-character variable to receive the form type of the printer form used when printing a job in the System/36 environment. A value of *STD or a user-defined form type is returned in the variable.

CL var for DFTMSGACN (9) (DFTMSGACN)

Specifies the name of a 9-character variable to receive the default message action used by the System/36 environment when an error occurs during the running of a CL command within a System/36 environment procedure. A value of *CONTINUE, *HALT, IGNORE, or *CANCEL is returned in the variable.

CL var for HALTOPT (4) (HALTOPT)

Specifies the name of a 4-character variable to receive the options list for continuation after an error occurs in the System/36 environment and *HALT is specified for the default message action.

CL var for EVKJOBINIT (6) (EVKJOBINIT)

Specifies the name of a 6-character variable to receive the value for the method used to start EVOKE jobs or job steps in the System/36 environment. A value of *IMMED or *JOBQ is returned in the variable.

CL var for EVKJOBPOL (8) (EVKJOBPOL)

Specifies the name of an 8-character variable to receive the value for the storage pool used for jobs started with the *IMMED option in the System/36 environment. A value of *BASE or *CURRENT is returned in the variable.

CL var for EVKJOBPTY (10) (EVKJOBPTY)

Specifies the name of a 10-character variable to receive the value for the priority at which a job is started when it is started with the *IMMED option in the System/36 environment. A value ranging from 1 through 99 or the value *SUBMITTER is returned in the variable.

CL var for SRCRCDLEN (3) (SRCRCDLEN)

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.

CL var for CHGACT (4) (CHGACT)

Specifies the name of a 4-character CL variable to receive the value indicating whether the configuration object can be updated using the Change System/36 (CHGS36) command while others are signed on to the System/36 environment. A value of *NO or *YES is returned in the variable.

CL var for ADDS36ONLY (4) (ADDS36ONLY)

Specifies the name of a 4-character CL variable to receive the value indicating whether workstation devices can be added to the System/36 environment configuration when the device signs on to the System/36 environment. A value of *NO or *YES is returned in the variable.

CL var for ICFSUBST (4) (ICFSUBST)

Specifies the name of a 4-character CL variable to receive the value indicating whether to scan intersystem communications function (ICF) start requests for substitution expressions. A value of *NO or *YES is returned in the variable.

CL var for MRTUSRPRF (8) (MRTUSRPRF)

Specifies the name of an 8-character variable to receive the user profile under which the Multiple Requester Terminal (MRT) program is running. A value of *OWNER or *FRSTUSR is returned in the variable.

CL var for MRTAUT (8) (MRTAUT)

Specifies the name of an 8-character variable to receive the user authority to files used by the MRT program. A value of *ALLUSR or *FRSTUSR is returned in the variable.

CL var for MRTDLY (5) (MRTDLY)

Specifies the name of a 5-character variable to receive the time in seconds that the system delays before ending the MRT program. A value ranging from 0 through 32767 is returned in the variable.

CL var for MRTJOBINIT (6) (MRTJOBINIT)

Specifies the name of a 6-character variable to receive the value for the method used to start an MRT job in the System/36 environment. A value of *IMMED or *JOBQ is returned in the variable.

CL var for MRTJOBPOL (8) (MRTJOBPOL)

Specifies the name of an 8-character variable to receive the value for the storage pool to be used for an MRT job started with the *IMMED option in the System/36 environment. A value of *BASE or *CURRENT is returned in the variable.

CL var for MRTJOBPTY (10) (MRTJOBPTY)

Specifies the name of a 10-character variable to receive the the value for the priority to start an MRT job started with the *IMMED option. A value ranging from 1 through 99 or the value *SUBMITTER is returned in the variable.

Examples

```
RTVS36A   RCBBLK(&RBLOCK)   HALT(&OPTION)   MRTUSRPRF(&USERID)
```

This command retrieves the shared file record blocking value, the halt options list, and the user profile under which the MRT is running. The file record blocking value is copied into the CL variable &RBLOCK, which must be 4 characters in length. The halt options list is copied into the CL variable &OPTION, which must be at least 4 characters in length. The user profile under which the MRT is running is copied into the CL variable &USERID, which must be 8 characters in length.

Error messages

None

Retrieve Service Agent (RTVSRVAGT)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve Service Agent (RTVSRVAGT) command allows a user to retrieve information about Service Agent.

Up to three valid days of the week for which Service Agent automatic processing can occur may be retrieved and placed in specified CL variable(s). The days of the week are valid for the system or logical partition on which the command is used.

Parameters

Keyword	Description	Choices	Notes
DAY1	CL variable for day 1	*Character value*	Optional
DAY2	CL variable for day 2	*Character value*	Optional
DAY3	CL variable for day 3	*Character value*	Optional

CL variable for day 1 (DAY1)

Specifies the name of the CL variable that receives the first day of the week for which Service Agent automatic processing can occur. The variable must be a character variable with a minimum length of 4 characters.

The returned day of the week is one of the following special values: *MON, *TUE, *WED, *THU, *FRI, *SAT, or *SUN.

CL variable for day 2 (DAY2)

Specifies the name of the CL variable that receives the second day of the week for which Service Agent automatic processing can occur. The variable must be a character variable with a minimum length of 4 characters.

The returned day of the week is one of the following special values: *MON, *TUE, *WED, *THU, *FRI, *SAT, or *SUN.

CL variable for day 3 (DAY3)

Specifies the name of the CL variable that receives the third day of the week for which Service Agent automatic processing can occur. The variable must be a character variable with a minimum length of 4 characters.

The returned day of the week is one of the following special values: *MON, *TUE, *WED, *THU, *FRI, *SAT, or *SUN.

Examples

```
DCL  &FIRSTDAY   TYPE(*CHAR) LEN(4)
DCL  &SECONDDAY  TYPE(*CHAR) LEN(4)
DCL  &THIRDDAY   TYPE(*CHAR) LEN(4)
 :
RTVSRVAGT   DAY1(&FIRSTDAY)  DAY2(*SECONDDAY)  DAY3(&THIRDDAY)
```

This command retrieves the three valid days of the Service Agent automatic functions (Auto PTF and Auto test).

Error messages

*ESCAPE Messages

CPF9899

    Error occurred during processing of command.

Retrieve Stop Word List Source (RTVSWLSRC)

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

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.

Parameters

Keyword	Description	Choices	Notes
LANGID	Language ID	*Character value*	Required, Positional 1
SRCFILE	Source file	*Qualified object name*	Required, Positional 2
	Qualifier 1: Source file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***LANGID**	Optional
TYPE	Stop word list type	***IBM**, *USER	Optional

Language ID (LANGID)

Specifies the language identifier (ID) for the stop word list.

This is a required parameter.

Source file (SRCFILE)

Specifies the qualified name of the source file used to receive the stop word list words. The contents of the source file are replaced.

source-file-name
> Specify the name of the source file.

The possible library values are:

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

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

library-name
> Specify the name of the library where the source file is located.

This is a required parameter.

© Copyright IBM Corp. 1998, 2006

217

Source member (SRCMBR)

Specifies the name of the source file member used to receive the stop word list words. The member is in the source file specified on the SRCFILE parameter.

*LANGID
> The language ID is used as the source file member name.

source-file-member-name
> Specify the name of the member in the source file used to receive the stop word list.

Stop word list type (TYPE)

Specifies the type of stop word list being retrieved

*IBM   The stop word list is IBM-supplied.

*USER
> The stop word list is user-created.

Examples

```
RTVSWLSRC   LANGID(ENG)   SRCFILE(MYLIB/MYFILE)
```

This command retrieves the stop word list into source file MYFILE in library MYLIB that has the language ID ENG.

Error messages

*ESCAPE Messages

CPF8723
> Record length too small for database source file.

CPF8725
> &1 type stop word list not supported for language.

CPF9899
> Error occurred during processing of command.

Retrieve System Information (RTVSYSINF)

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

The Retrieve System Information (RTVSYSINF) command gathers key system information from your system and places it in the library specified for the Library (LIB) parameter.

Restrictions: To use this command, you must have *SAVSYS authority.

Parameters

Keyword	Description	Choices	Notes
LIB	Library	*Name*	Required, Positional 1

Library (LIB)

Specifies the library in which objects containing system information are to be stored. There is no default value, and the library must exist on the system.

library
    Specify the library in which objects containing system information are to be stored. The library must exist on the system.

Examples

None

Error messages

<u>*ESCAPE Messages</u>

CPF2110
    Library &1 not found.

CPF222E
    &1 special authority is required.

CPFA95F
    Errors occurred while retrieving system information.

Retrieve System Value (RTVSYSVAL)

The Retrieve System Value (RTVSYSVAL) command is used in a CL program to retrieve the value from the specified system value so that it can be used in the program. The value is returned (copied) to the specified CL variable in the program.

Restrictions:

1. This command is valid only in CL programs or REXX procedures.
2. The attributes of the system value and the receiving CL variable must be compatible.

Parameters

Keyword	Description	Choices	Notes
SYSVAL	System value	QABNORMSW, QACGLVL, QACTJOB, QADLACTJ, QADLSPLA, QADLTOTJ, QALWJOBITP, QALWOBJRST, QALWUSRDMN, QASTLVL, QATNPGM, QAUDCTL, QAUDENDACN, QAUDFRCLVL, QAUDLVL, QAUDLVL2, QAUTOCFG, QAUTORMT, QAUTOSPRPT, QAUTOVRT, QBASACTLVL, QBASPOOL, QBOOKPATH, QCCSID, QCENTURY, QCFGMSGQ, QCHRID, QCHRIDCTL, QCMNARB, QCMNRCYLMT, QCNTRYID, QCONSOLE, QCRTAUT, QCRTOBJAUD, QCTLSBSD, QCURSYM, QDATE, QDATETIME, QDATFMT, QDATSEP, QDAY, QDAYOFWEEK, QDBFSTCCOL, QDBRCVYWT, QDECFMT, QDEVNAMING, QDEVRCYACN, QDSCJOBITV, QDSPSGNINF, QDYNPTYADJ, QDYNPTYSCD, QENDJOBLMT, QFRCCVNRST, QHOUR, QHSTLOGSIZ, QIGC, QIGCCDEFNT, QIGCFNTSIZ, QINACTMSGQ, QINACTITV, QIPLDATTIM, QIPLSTS, QIPLTYPE, QJOBMSGQFL, QJOBMSGQMX, QJOBMSGQSZ, QJOBMSGQTL, QJOBSPLA, QKBDBUF, QKBDTYPE, QLANGID, QLEAPADJ, QLIBLCKLVL, QLMTDEVSSN, QLMTSECOFR, QLOCALE, QLOGOUTPUT, QMAXACTLVL, QMAXJOB, QMAXSGNACN, QMAXSIGN, QMAXSPLF, QMCHPOOL, QMINUTE, QMLTTHDACN, QMODEL, QMONTH, QPASTHRSVR, QPFRADJ, QPRBFTR, QPRBHLDITV, QPRCFEAT, QPRCMLTTSK, QPRTDEV, QPRTKEYFMT, QPRTTXT, QPWDEXPITV, QPWDLMTAJC, QPWDLMTCHR, QPWDLMTREP, QPWDLVL, QPWDMAXLEN, QPWDMINLEN, QPWDPOSDIF, QPWDRQDDGT, QPWDRQDDIF, QPWDVLDPGM, QPWRDWNLMT, QPWRRSTIPL, QQRYDEGREE, QQRYTIMLMT, QRCLSPLSTG, QRETSVRSEC, QRMTIPL, QRMTSIGN, QRMTSRVATR, QSAVACCPTH, QSCANFS, QSCANFSCTL, QSCPFCONS, QSECOND, QSECURITY, QSETJOBATR, QSFWERRLOG, QSHRMEMCTL, QSPCENV, QSPLFACN, QSRLNBR, QSRTSEQ, QSRVDMP, QSTGLOWACN, QSTGLOWLMT, QSTRPRTWTR, QSTRUPPGM, QSTSMSG, QSVRAUTITV, QSYSLIBL, QTHDRSCADJ, QTHDRSCAFN, QTIMADJ, QTIME, QTIMSEP, QTIMZON, QTOTJOB, QTSEPOOL, QUPSDLYTIM, QUPSMSGQ, QUSEADPAUT, QUSRLIBL, QUTCOFFSET, QVFYOBJRST, QYEAR	Required, Positional 1
RTNVAR	CL variable for returned value	*Not restricted*	Required, Positional 2

System value (SYSVAL)

Specifies the name of the system value whose value is retrieved and returned for use in the program. The names and descriptions of the system values that can be specified are in the Work Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

This is a required parameter.

The system values are:

QABNORMSW
       Previous end of system indicator. This value cannot be changed.

- '0' means previous end was normal.
- '1' means previous end was abnormal.

QACGLVL

Accounting level. Changes made to this system value take effect for jobs started after the change is made.

- *NONE - No accounting information is written to a journal.
- *JOB - Job resource use is written to a journal.
- *PRINT - Spooled and printer file resource use is written to a journal.

QACTJOB

Initial number of active jobs for which storage is allocated. Changes made to this system value take effect at the next IPL.

QADLACTJ

Additional number of active jobs for which storage is allocated. Changes made to this system value take effect immediately.

QADLSPLA

Additional storage for extending spooling control block (bytes). The operating system no longer uses this system value. Changes made to this system value have no effect.

QADLTOTJ

Additional total number of jobs for which storage is allocated. Changes made to this system value take effect immediately.

QALWJOBITP

Allow jobs to be interrupted. This system value specifies how the system responds to user initiated requests to interrupt a job to run a user-defined exit program in that job. The Call Job Interrupt Program (QWCJBITP) API in the iSeries Information Center contains information on using job interrupt exit programs. The Change Job Interrupt Status (QWCCJITP) API in the iSeries Information Center contains information on retrieving and changing the interrupt status of a job. The interrupt status of an active job can be changed at any time but will only take effect when the value of QALWJOBITP allows jobs to be interrupted. Changes made to this system value take effect immediately. The shipped value is 0.

- 0 means the system will not allow jobs to be interrupted to run user-defined exit programs. All new jobs becoming active will default to be uninterruptible. All active jobs are uninterruptible regardless of what the job interrupt status is set to.
- 1 means the system will allow jobs to be interrupted to run user-defined exit programs. All new jobs becoming active will default to be uninterruptible.
- 2 means the system will allow jobs to be interrupted to run user-defined exit programs. All new jobs becoming active will default to be interruptible.

QALWOBJRST

Allow object to be restored. This system value determines whether objects with security-sensitive attributes are restored. See Restore options for additional information.

QALWUSRDMN

Allow user domain objects in libraries or directories. This system value specifies which libraries on the system can contain the user domain user objects *USRSPC (user space), *USRIDX (user index), and *USRQ (user queue). Changes made to this system value take effect immediately.

QASTLVL

Assistance level. Indicates the Operational Assistant level of system displays for user profiles where ASTLVL(*SYSVAL) is specified. Changes made to this system value take effect immediately.

- *BASIC - The Operational Assistant user interface is used.
- *INTERMED - The system interface is used.

- *ADVANCED - The expert system interface is used.

If a command does not have an *ADVANCED level interface, *INTERMED is used.

QATNPGM

Attention program. If *ASSIST is specified for this system value, the Operational Assistant main menu is called when the user presses the Attention (Attn) key. This value can be changed to the name of a program, which will be called when the user presses the Attn key in a job where ATNPGM(*SYSVAL) is specified in the user profile. Changes made to this system value take effect immediately.

QAUDCTL

Audit control. This system value contains the on and off switches for object and user action auditing. This system value activates auditing on the system that is selected by the Change Object Auditing (CHGOBJAUD) and Change User Auditing (CHGUSRAUD) commands and the QAUDLVL and QAUDLVL2 system values. Changes made to this system value take effect immediately.

- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL).
- *NONE - No security auditing is done on the system. This is the shipped value.
- *AUDLVL - The actions specified in the QAUDLVL and QAUDLVL2 system values will be logged to the security journal. Also actions specified by a user profile's action auditing values will be audited. A user profile's action auditing values are set through the AUDLVL parameter on the CHGUSRAUD command.
- *OBJAUD - Actions against objects that have an object audit value other than *NONE will be audited. An object's audit value is set through the Change Auditing Value (CHGAUD) command or the CHGOBJAUD command.
- *NOQTEMP - No auditing of most objects in QTEMP is done. You must specify *NOQTEMP with either *OBJAUD or *AUDLVL. You can not specify *NOQTEMP by itself.

QAUDENDACN

Audit journal error action. This system value specifies the action to be taken by the system if errors occur when an audit journal entry is being sent by the operating system to the security audit journal. Changes made to this system value take effect immediately.

- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL).
- *NOTIFY - Notification of failure is sent to the QSYSOPR and QSYSMSG message queues, and then the action that caused the audit attempt continues.
- *PWRDWNSYS - The Power Down System (PWRDWNSYS) command is issued. The system will then be brought up in a restricted state on the following IPL, and then only a user with audit (*AUDIT) and all object (*ALLOBJ) special authority can sign on the system.

QAUDFRCLVL

Force audit journal. This system value specifies the number of audit journal entries that can be written to the security auditing journal before the journal entry data is forced to auxiliary storage.

- 1 through 100.
- *SYS - The system determines when the journal entries are to be written to auxiliary storage based on internal system processing. *SYS cannot be returned in a decimal variable, so the command returns 0 when the value *SYS is specified.
- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL). *NOTAVL cannot be returned in a decimal variable, so the command returns -1 in place of *NOTAVL.

Changes made to this system value take effect immediately.

QAUDLVL

Security auditing level. Controls the level of action auditing on the system. Changes made to this system value take effect immediately for all jobs running on the system.

- *NONE - No security action auditing will occur on the system. This is the shipped value.
- *AUDLVL2 - Both QAUDLVL and QAUDLVL2 system values will be used to determine the security actions to be audited.

  Note:
  - If you wish to use the QAUDLVL2 system value exclusively, set the QAUDLVL system value to *AUDLVL2 and add your auditing values to the QAUDLVL2 system value.
  - If you wish to use both system values you can set your values in the QAUDLVL system value along with the *AUDLVL2 value, then add any additional values to the QAUDLVL2 system value.

- *ATNEVT - Attention events are audited. Attention events are conditions that require further evaluation to determine the condition's security significance. For example, intrusion monitor events need to be examined to determine whether the condition is an intrusion or a false positive.
- *AUTFAIL - Authorization failures are audited.
- *CREATE - All object creations are audited. Objects created into library QTEMP are not audited.
- *DELETE - All deletions of external objects on the system are audited. Objects deleted from library QTEMP are not audited.
- *JOBDTA - Actions that affect a job are audited.
- *NETBAS - Network base functions are audited.
- *NETCLU - Cluster and cluster resource group operations are audited.
- *NETCMN - Networking and communications functions are audited.

  Note: *NETCMN is composed of several values to allow you to better customize your auditing. If you specify all of the values, you will get the same auditing as if you specified *NETCMN. The following values make up *NETCMN.
  - *NETBAS
  - *NETCLU
  - *NETFAIL
  - *NETSCK

- *NETFAIL - Network failures are audited.
- *NETSCK - Socket tasks are audited.
- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL).
- *OBJMGT - Generic object tasks are audited.
- *OFCSRV - OfficeVision tasks are audited.
- *OPTICAL - All optical functions are audited.
- *PGMADP - Adopting authority from a program owner is audited.
- *PGMFAIL - Program failures are audited.
- *PRTDTA - Printing functions are audited.
- *SAVRST - Save and restore information is audited.
- *SECCFG - Security configuration is audited.
- *SECDIRSRV- Changes or updates when doing directory service functions are audited.
- *SECIPC - Changes to interprocess communications are audited.
- *SECNAS - Network authentication service actions are audited.
- *SECRUN - Security run time functions are audited.

- *SECSCKD - Socket descriptors are audited.
- *SECURITY - All security-related functions are audited.

  Note: *SECURITY is composed of several values to allow you to better customize your auditing. If you specify all of the values, you will get the same auditing as if you specified *SECURITY. The following values make up *SECURITY.
    - *SECCFG
    - *SECDIRSRV
    - *SECIPC
    - *SECNAS
    - *SECRUN
    - *SECSCKD
    - *SECVFY
    - *SECVLDL
- *SECVFY - Use of verification functions are audited.
- *SECVLDL - Changes to validation list objects are audited.
- *SERVICE - For a list of all the service commands and API calls that are audited, see the iSeries Security Reference publication
- *SPLFDTA - Spooled file functions are audited.
- *SYSMGT - System management tasks are audited.

QAUDLVL2

Security auditing level extension. This system value is required when more than sixteen auditing values are needed. Specifying *AUDLVL2 as one of the values in the QAUDLVL system value will cause the system to also look for auditing values in the QAUDLVL2 system value. Changes made to this system value take effect immediately for all jobs running on the system.

- *NONE - No auditing values are contained in this system value. This is the shipped value.
- *ATNEVT - Attention events are audited. Attention events are conditions that require further evaluation to determine the condition's security significance. For example, intrusion monitor events need to be examined to determine whether the condition is an intrusion or a false positive.
- *AUTFAIL - Authorization failures are audited.
- *CREATE - All object creations are audited. Objects created into library QTEMP are not audited.
- *DELETE - All deletions of external objects on the system are audited. Objects deleted from library QTEMP are not audited.
- *JOBDTA - Actions that affect a job are audited.
- *NETBAS - Network base functions are audited.
- *NETCLU - Cluster and cluster resource group operations are audited.
- *NETCMN - Networking and communications functions are audited.

  Note: *NETCMN is composed of several values to allow you to better customize your auditing. If you specify all of the values, you will get the same auditing as if you specified *NETCMN. The following values make up *NETCMN.
    - *NETBAS
    - *NETCLU
    - *NETFAIL
    - *NETSCK
- *NETFAIL - Network failures are audited.
- *NETSCK - Socket tasks are audited.

- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL).
- *OBJMGT - Generic object tasks are audited.
- *OFCSRV - OfficeVision tasks are audited.
- *OPTICAL - All optical functions are audited.
- *PGMADP - Adopting authority from a program owner is audited.
- *PGMFAIL - Program failures are audited.
- *PRTDTA - Printing functions are audited.
- *SAVRST - Save and restore information is audited.
- *SECCFG - Security configuration is audited.
- *SECDIRSRV- Changes or updates when doing directory service functions are audited.
- *SECIPC - Changes to interprocess communications are audited.
- *SECNAS - Network authentication service actions are audited.
- *SECRUN - Security run time functions are audited.
- *SECSCKD - Socket descriptors are audited.
- *SECURITY - All security-related functions are audited.

  Note: *SECURITY is composed of several values to allow you to better customize your auditing. If you specify all of the values, you will get the same auditing as if you specified *SECURITY. The following values make up *SECURITY.
  - *SECCFG
  - *SECDIRSRV
  - *SECIPC
  - *SECNAS
  - *SECRUN
  - *SECSCKD
  - *SECVFY
  - *SECVLDL
- *SECVFY - Use of verification functions are audited.
- *SECVLDL - Changes to validation list objects are audited.
- *SERVICE - For a list of all the service commands and API calls that are audited, see the iSeries Security Reference publication
- *SPLFDTA - Spooled file functions are audited.
- *SYSMGT - System management tasks are audited.

QAUTOCFG

Automatic device configuration indicator. Changes made to this system value take effect immediately.
- 0 means auto-configuration is off.
- 1 means auto-configuration is on.

QAUTOSPRPT

Automatic system disabled reporting. The operating system no longer uses this system value. Changes made to this system value have no effect.

QAUTORMT

Automatic configuration for remote controllers. The QAUTORMT system value controls the automatic configuration of remote controllers.
- 0 means auto-configuration is off.
- 1 means auto-configuration is on.

QAUTOVRT

Automatic virtual device configuration indicator. The user must have *ALLOBJ authority to change this system value. Changes made to this system value take effect immediately. See Autoconfigure virtual devices for additional information.

QBASACTLVL

Activity level of base storage pool. Changes made to this system value take effect immediately.

QBASPOOL

Minimum size of base storage pool (in Kilobytes). Changes made to this system value take effect immediately.

QBOOKPATH

Book and bookshelf search path. The operating system no longer uses this system value. Changes made to this system value have no effect.

QCCSID

Coded character set identifier. Changes made to this system value take effect for jobs started after the change is made.

QCENTURY

Century value for the system date.

- 0 indicated years 19**XX**.

- 1 indicates years 20**XX**.

QCFGMSGQ

Configuration message queue used to specify the message queue to receive communication messages. Both an object name and library name can be specified. A change to this system value takes effect when a line, controller, or device description that supports the MSGQ parameter is varied on.

QCHRID

Default graphic character set and code page used for displaying or printing data. Changes made to this system value take effect for display files, display device descriptions, and printer files that are created, changed, or overridden after the change.

QCHRIDCTL

Character identifier control for the job. This attribute controls the type of CCSID conversion that occurs for display files, printer files, and panel groups. The *CHRIDCTL special value must be specified for the CHRID parameter on the create, change, or override commands for display files, printer files, and panel groups before this attribute is used.

- 0 means the *DEVD special value is used.

- 1 means the *JOBCCSID special value is used.

QCMNARB

Communication arbiters. The number of communication arbiter jobs that are available to process work for controllers and devices. A change to this value takes effect on the next IPL. The shipped value is *CALC.

- *CALC: The operating system calculates the number of communication arbiter jobs.

- 0 - 99: Specifies the number of communication arbiter jobs that are available to process work for controllers and devices.

Note: If this system value is set to zero (0), the work in these jobs is done in QSYSARB and QLUS system jobs as opposed to the communication arbiters.

QCMNRCYLMT

Provides recovery limits for system communications recovery. Specifies the number of recovery attempts to make and when an inquiry message is sent to the device message queue or to the

system operator when the specified number of recovery attempts have been reached. Changes made to this system value do not affect a currently varied on device, but is in effect when a device is varied on after the change.

QCNTRYID

Default country or region identifier. Changes to this system value take effect for jobs started after the change is made.

QCONSOLE

System console. This value is not changeable.

QCRTAUT

Public authority for created objects. You must have *ALLOBJ and *SECADM special authorities to change this system value. Changes made to this system value take effect immediately.

- *CHANGE means the user can change the object and perform basic functions on the object. Change authority allows the user to perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. Change authority provides object operational authority and all data authority.

- *ALL means the user can control the object's existence, specify the security for the object, change the object, change the owner for the object, and perform basic functions on the object. All authority allows the user to perform all operations on the object except those limited to the owner or controlled by authorization list management rights. If the object is an authorization list, the user cannot add, change, or remove users, or transfer ownership of the authorization list.

- *USE means the user can perform basic operations on the object, such as run a program or read a file. The user is prevented from changing the object. Use authority provides object operational authority and read authority.

- *EXCLUDE authority prevents the user from accessing the object.

QCRTOBJAUD

Create object auditing. This system value specifies the default object auditing value for an object created into a library or directory. The object auditing value determines whether an audit journal entry is sent to the system auditing journal when an object is used or changed. Changes made to this system value take effect immediately.

- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL).

- *NONE - No auditing entries are sent for the object.

- *USRPRF - Auditing entries are sent if the user is currently being audited.

- *CHANGE - Auditing entries are sent if the object is changed.

- *ALL - Auditing entries are sent if the object is used or changed.

QCTLSBSD

Controlling subsystem description name. Both an object name and library name can be specified. Changes made to this system value take effect at the next IPL.

QCURSYM

Currency symbol. Changes made to this system value take effect immediately.

QDATE

System date. Changes made to this system value take effect immediately.

QDATETIME

System date and time. This is the date and time for the local system time as a single value. Retrieving or changing this value is similar to retrieving or changing QDATE and QTIME in a single operation. The format of the field is YYYYMMDDHHNNSSXXXXXX where YYYY is the year, MM is the month, DD is the day, HH is the hours, NN is the minutes, SS is the seconds, and XXXXXX is the microseconds. Changes made to this system value take effect immediately.

QDATFMT

Date format. Changes made to this system value take effect for jobs started after the change is made.

QDATSEP

Date separator. Changes made to this system value take effect for jobs started after the change is made.

QDAY Day of the month (day of the year if the system date format is Julian). Changes made to this system value take effect immediately.

QDAYOFWEEK

The day of the week.

- *SUN - Sunday
- *MON - Monday
- *TUE - Tuesday
- *WED - Wednesday
- *THU - Thursday
- *FRI - Friday
- *SAT - Saturday

QDBFSTCCOL

Database file statistics collection. Specifies the type of statistics collection requests that are allowed to be processed in the background by system job, QDBFSTCCOL. Changes made to this system value take effect immediately.

- *ALL means all user requested database file statistics collection requests and statistics collections automatically requested by the database manager are allowed to be processed by the database statistics system job.
- *SYSTEM means only automatically requested database statistics collection requests by the database manager are allowed to be processed by the database statistics system job.
- *USER means only user requested database file statistics collection requests are allowed to be processed by the database statistics system job.
- *NONE means no database file statistics collection requests are allowed to be processed by the database statistics system job.

QDBRCVYWT

Database recovery wait indicator. Changes to this system value take effect at the next IPL in unattended mode.

- 0 means do not wait.
- 1 means wait.

QDECFMT

Decimal format. Changes made to this system value take effect immediately.

QDEVNAMING

Indicates the device naming convention. Changes made to this system value take effect the next time a device is automatically configured. Existing configured device names are not changed.

- *NORMAL means follow iSeries standards.
- *S36 means follow S/36 standards.
- *DEVADR means device names are derived from resource names.

QDEVRCYACN

Specifies the action taken when an I/O error occurs for the job's requesting program device. Changes made to this system value take effect for jobs started after the change is made.

- *DSCMSG disconnects the job. On reconnection, an error message will be sent to the user's application program.

- *DSCENDRQS disconnects the job. On reconnection, a cancel request function should be performed to return control of the job back to the last request level.
- *ENDJOB ends the job. A job log will be produced for the job. A message will be sent to the job log and to the QHST log indicating that the job was ended because of device error.
- *ENDJOBNOLIST ends the job. A job log will not be produced for the job. A message will be sent to the QHST log indicating that the job was ended because of device error.
- *MSG signals the I/O error message to the application program. The application program performs error recovery itself.

QDSCJOBITV

Time interval that a job can be disconnected before it is ended. Changes made to this system value take effect immediately. An interactive job can be disconnected with the Disconnect Job (DSCJOB) command when it has been inactive for an interval of time (the system values QINACTIV and QINACTMSGQ), or when an Input/Output error occurs at the interactive job's work station (the system value QDEVRCYACN).

- 5-1440 is the time out interval in minutes.
- *NONE means no time out interval.

QDSPSGNINF

Controls the display of sign-on information. Changes made to this system value take effect immediately.

- 0 means the sign-on information is not displayed.
- 1 means the sign-on information is displayed.

QDYNPTYADJ

Dynamic priority adjustment. The QDYNPTYADJ system value controls whether the priority of interactive jobs is dynamically adjusted to maintain high performance of batch job processing. This adjustment capability is only effective on systems that are rated for both interactive and non-interactive throughput and have Dynamic Priority Scheduling enabled. A change to this value takes effect at the next IPL.

- 0 means the dynamic priority adjustment support is turned off.
- 1 means the dynamic priority adjustment support is turned on.

QDYNPTYSCD

Dynamic priority scheduler. The QDYNPTYSCD system value controls the dynamic priority scheduler algorithm. The value allows the use of dynamic priority scheduling.

- 0 means the dynamic priority scheduler is off.
- 1 means the dynamic priority scheduler is on.

QENDJOBLMT

Maximum time (in seconds) for application clean up during immediate ending of a job.

When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job. When the signal handling procedure for the SIGTERM signal is given control, the procedure can take the appropriate actions to avoid undesirable results such as application data that has been partially updated. If the SIGTERM signal handler has not completed in the specified time, the system ends the job.

When the job is ended in a controlled manner, the maximum time for the SIGTERM signal handler is specified on the command. When the job is ended in an immediate manner, the maximum time for the SIGTERM signal handler is specified by this system value. This time limit is used when ending one job, when ending all the jobs in a subsystem, or when ending all jobs in all subsystems. After two minutes, the system operator can use the End Job (ENDJOB) command with OPTION(*IMMED) to override the QENDJOBLMT value and end individual jobs immediately.

A change to this value takes effect immediately. Jobs that are already ending are not affected.

QFRCCVNRST

Force conversion on restore. This system value allows you to specify whether or not to convert programs, service programs, SQL packages, and module objects during the restore. It can also prevent some objects from being restored. The default value on the restore commands use the value of this system value. Changes to this system value will take effect immediately.

0 Do not convert anything. Do not prevent anything from being restored.

1 Objects with validation errors will be converted.

2 Objects requiring conversion to be used on the current version of the operating system and objects with validation errors will be converted.

3 Objects suspected of having been tampered with, objects containing validation errors, and objects requiring conversion to be used by the current version of the operating system will be converted.

4 Objects that contain sufficient creation data to be converted and do not have valid digital signatures will be converted. An object that does not contain sufficient creation data will be restored without conversion. NOTE: Objects (signed and unsigned) that have validation errors, are suspected of having been tampered with, or require conversion to be used by the current version of the operating system, but cannot be converted will not be restored.

5 Objects that contain sufficient creation data will be converted. An object that does not contain sufficient creation data will be restored. NOTE: Objects that have validation errors, are suspected of having been tampered with, or require conversion to be used on the current version of the operating system, but cannot be converted will not be restored.

6 All objects that do not have a valid digital signature will be converted. NOTE: An object with a valid digital signature that also has a validation error, is suspected of having been tampered with, or requires conversion to be used on the current version of the operating system, but cannot be converted will not be restored.

7 Every object will be converted.

When an object is converted, its digital signature is discarded. The state of the converted object is set to user state. After conversion, objects will have a good validation value and are not suspected of having been tampered with.

QHOUR

Hour of the day. Changes made to this system value take effect immediately.

QHSTLOGSIZ

Maximum number of records for each version of the history log. Valid values range from 1 to 10,000,000 or the special value *DAILY which means that a new version of the history log is created each time the date in the history log messages changes, or when the current log version reaches the maximum size of 10,000,000 records. *DAILY cannot be returned in a decimal variable, so the Retrieve System Value (RTVSYSVAL) command returns a value of -1 when the system value is set to *DAILY. Specifying a value of -1 on the Change System Value (CHGSYSVAL) command has the same effect as specifying *DAILY. Changes made to this system value take effect when the next version of the history log is created.

QIGC Indicates whether the double-byte character set (DBCS) version of the system is installed. This value cannot be changed.

- 0 means the DBCS version is not installed.
- 1 means the DBCS version is installed.

QIGCCDEFNT

Double byte character set (DBCS) coded font name. Used when transforming an SNA character

string (SCS) into an Advanced Function Printing data stream (AFPDS) and when creating an AFPDS spooled file with shift in/shift out (SI/SO) characters in the data. Changes made to this system value take effect immediately.

QIGCFNTSIZ

Double byte coded font point size. Used along with the system value, QIGCCDEFNT, double byte coded font. They will be used when transforming SNA character string (SCS) into an Advanced Function Printing Data Stream (AFPDS) and when creating an AFPDS spooled file with shift in/ shift out (SI/SO) characters present in the data.

- *NONE means that no point size is identified to the system. The point size is selected by the system based on the type of printer used.
- 000.1 - 999.9 means the point size for the double byte coded font.

QINACTITV

Inactive interactive job time out interval in minutes. When the time interval is changed to a value other than *NONE a new inactivity interval is established and the analysis of job inactivity is started again. The system value QINACTMSGQ determines the action the system takes. For information on enforcement for target pass-through and TELNET sessions, see the Work Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter

Local jobs that are currently signed on to a remote system are excluded. For example, a work station is directly attached to system A, and system A has QINACTIV set on. If Display Station Pass-through or TELNET is used to sign on to system B, this work station is not affected by the QINACTITV value set on system A. Changes made to this system value take effect immediately.

- *NONE means that the system does not check for inactivity.
- 5 - 300 means the number of minutes a job can be inactive before action is taken.

QINACTMSGQ

The qualified name of a message queue to which job inactive messages will be sent if QINACTMSGQ is not *NONE. The message queue must exist before the system value can be changed to a message queue name. Both an object name and library name can be specified. Changes made to this system value take effect immediately.

- *ENDJOB means that interactive jobs, secondary jobs, and group jobs will be ended.
- *DSCJOB means that interactive jobs, secondary jobs, and group jobs will be disconnected.
- Message queue name is the name of a message queue that receives a message when a job has been inactive.

QIPLDATTIM

Date and time for automatic IPL. This system value can be set independently in each partition. If the primary partition is powered down at the time an automatic IPL should occur in a secondary partition, the IPL will not occur. When the primary partition does IPL, the secondary partition will be IPLed if its IPL date and time is past due. The secondary partition will not IPL if it was configured with an IPL action of hold. Changes made to this system value take effect immediately.

QIPLDATTIM is a single system value with two parts:

- Date: The date an IPL automatically occurs on the system. The date is specified in QDATFMT format with no date separators.
- Time: The time an IPL automatically occurs on the system. The time is specified with no time separators.

*NONE, which indicates that no timed automatic IPL is scheduled, can be specified instead of a specific date and time.

The following example shows how to change the IPL date and time to September 10, 1993 (QDATFMT is MDY) at 9:00 a.m.

```
CHGSYSVAL SYSVAL(QIPLDATTIM) VALUE('091093 090000')
```

QIPLSTS

Initial program load (IPL) status indicator.

- 0 means operator panel IPL.
- 1 means auto-IPL after power restored.
- 2 means restart IPL.
- 3 means time of day IPL.
- 4 means remote IPL.

QIPLTYPE

Indicates the type of IPL to perform. Changes made to this system value take effect at the next manual IPL.

- 0 means unattended IPL.
- 1 means attended IPL with dedicated service tools.
- 2 means attended IPL with console in debug mode.

    Note: You should only use this for problem analysis because it prevents other devices on the work station controller from being used.

QJOBMSGQFL

Job message queue full action. This system value specifies how to handle the job message queue when it is considered full. Changes made to this system value take effect for jobs started after the change is made.

- *NOWRAP - The job message queue is not wrapped.
- *WRAP - The job message queue is wrapped.
- *PRTWRAP - The job message queue is wrapped and the messages that are being overlaid are printed.

QJOBMSGQMX

Job message queue maximum size. This system value specifies how large (in megabytes) a message queue can be before it is considered full. Changes made to this system value take effect for jobs started after the change is made.

QJOBMSGQSZ

Initial size of job message queue in kilobytes (KB). The operating system no longer uses this system value. Changes made to this system value have no effect.

QJOBMSGQTL

Maximum size of job message queue (in KB). The operating system no longer uses this system value. Changes made to this system value have no effect.

QJOBSPLA

Initial size of spooling control block for a job (in bytes). Changes made to this system value take effect when a cold start is requested during the installation of the operating system licensed program.

QKBDBUF

Keyboard buffer. Changes made to this system value take effect the next time someone logs on.

- *NO means turn off the type-ahead feature and the attention key buffering option.
- *TYPEAHEAD means turn on the type-ahead feature but turn off the attention key buffering option.
- *YES means turn on the type-ahead feature and the attention key buffering option.

QKBDTYPE

Keyboard language character set. Changes made to this system value take effect immediately.

QLANGID

Default language identifier. Changes to this system value take effect for jobs started after the change is made.

QLEAPADJ

Leap year adjustment. Changes made to this system value take effect immediately.

QLIBLCKLVL

Library locking level. Specifies whether libraries in a job's library search list are locked by that job. A change to this system value takes effect for all jobs that become active after the change.

- 0 means the libraries in a user job's library search list are not locked.
- 1 means the libraries in a user job's library search list are locked by that job.

QLMTDEVSSN

Limits concurrent device sessions. Changes made to this system value take effect immediately.

- 0 means you can sign on at multiple devices.
- 1 means you cannot sign on at more than one device.

QLMTSECOFR

Limit security officer device access. Changes made to this system value take effect immediately.

- 0 means users with *ALLOBJ or *SERVICE special authority can sign on any work station.
- 1 means users with *ALLOBJ or *SERVICE special authority must have explicit authority to a work station.

QLOCALE

Locale path name. This system value is used to set the locale for the system. The locale path name must be a path name that specifies a locale. A locale is made up of the language, territory, and code set combination used to identify a set of language conventions. The maximum path length allowed for the locale path name on the Change System Value (CHGSYSVAL) command is 1,024 bytes.

A change to this system value takes effect immediately. The shipped value may be different for different countries.

- *NONE means there is no locale path name for the QLOCALE system value.
- *C means the C locale is to be used.
- *POSIX means the POSIX locale is to be used.

QLOGOUTPUT

Job log output. This system value specifies how the job log will be produced when a job completes. This does not affect job logs produced when the message queue is full and the job message queue full action specifies *PRTWRAP. Messages in the job message queue are written to a spooled file, from which the job log can be printed, unless the Control Job Log Output (QMHCTLJL) API was used in the job to specify that the messages in the job log are to be written to a database file.

Changes made to this system value take effect immediately for jobs entering the system after the change is made.

- *JOBEND means the job log will be produced by the job itself. If the job cannot produce its own job log, the job log will be produced by a job log server.
- *JOBLOGSVR means the job log will be produced by a job log server.
- *PND means the job log will not be produced. The job log remains pending until removed.

QMAXACTLVL

Maximum activity level of the system. Changes made to this system value take effect immediately.

QMAXJOB

Maximum number of jobs that are allowed on the system. Changes made to this system value take effect immediately.

QMAXSGNACN

The system's response when the limit imposed by QMAXSIGN system value is reached. Changes made to this system value take effect the next time someone attempts to sign on the system.

- 1 means the device will be disabled.
- 2 means the user profile will be disabled.
- 3 means the device and the user profile will be disabled.

QMAXSIGN

Maximum number of not valid sign-on attempts allowed. Changes made to this system value take effect the next time someone attempts to sign on the system.

QMAXSPLF

Maximum number of spooled files that can be created per job. Changes made to this system value take effect immediately. Spooled files will not be deleted when this value is changed to a lower number. See the Printer Device Programming book for information on how this system value affects spooling for a job.

QMCHPOOL

Machine storage pool size (in KB). Changes made to this system value take effect immediately.

Note: Changes to the size of a pool may require pages to be written to auxiliary storage. The time required for the system to complete a large change may be greater than your default wait time. If this occurs, message CPF1001 (Wait time expired for system response.) is issued, even though the change completes.

QMINUTE

Minute of the hour. Changes made to this system value take effect immediately.

QMLTTHDACN

Multithreaded job action. This value controls the action to be taken when a function that may not be threadsafe is called in a multithreaded job. Changes made to this system value take effect immediately. The shipped value is 2.

- 1 means perform the function that is not threadsafe without sending a message.
- 2 means perform the function that is not threadsafe and send an informational message.
- 3 means do not perform the function that is not threadsafe.

QMODEL

System model number. The number or letters used to identify the model of the system. You cannot change QMODEL, but the 4-character value can be displayed or retrieved in user-written programs. The system model number system value is the same in each partition on a system.

QMONTH

Month of the year (not used for Julian dates). Changes made to this system value take effect immediately.

QPASTHRSVR

Pass-through servers. The number of target display station pass-through server jobs that are available to process display station pass-through, iSeries Access for Windows workstation function (WSF), and other 5250 emulation programs on programmable workstations. Changes made to this system value take effect immediately. The shipped value is *CALC.

QPFRADJ

Initial program load (IPL) performance adjustment and dynamic performance tuning. Dynamic performance tuning automatically changes storage pool sizes and activity levels for shared storage pools. Private storage pools are not changed. Changes made to this system value take effect immediately.

- 0 means no performance adjustment. Dynamic performance tuning is not started.
- 1 means performance adjustment at IPL. Dynamic performance tuning is not started.

- 2 means performance adjustment at IPL. Dynamic performance tuning is started. If QPFRADJ is changed from 2 to 0 or 1, dynamic performance tuning is stopped.
- 3 means dynamic performance tuning is started. If QPFRADJ is changed from 3 to 0 or 1, dynamic performance tuning is stopped.

If you create journal QPFRADJ in library QSYS, the dynamic tuning program keeps a record of the changes made to storage pool sizes, activity levels, and the performance level of the system when the changes were made (faulting rates per pool, pool sizes, and activity levels).

QPRBFTR

Problem filter name. Specifies the name of the filter object used by the service activity manager when processing problems. Changes to this system value take effect immediately.

QPRBHLDITV

Problem log entry hold interval. Changes made to this system value take effect immediately.

QPRCFEAT

Processor feature. The is the processor feature code level of the system. You cannot change QPRCFEAT, but the 4-character value can be displayed or retrieved in user-written programs. The processor feature system value is the same in each partition on a system.

QPRCMLTTSK

Processor multitasking. If the hardware on your system supports processor multitasking, this system value allows you to set the multitasking capability to be on, off, or System-controlled. If enabled, more than one set of task data will be resident in each CPU. Some workloads may experience increased performance due to caching implications. **Note:** The operating system will set the system value to 0 on the next IPL if it detects that the hardware does not support multitasking. Setting the value to system controlled will allow the system to manage the multitasking. Changes made to this system value take effect at the next IPL.

- 0 means that processor multitasking is turned off.
- 1 means that processor multitasking is turned on.
- 2 means that processor multitasking is under system control.

On some partitioned systems, this system value can only be changed from the primary partition.

For more information on partitions, see the Logical Partitions topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

QPRTDEV

Default printer device description. Changes made to this system value take effect for jobs started after the change is made.

QPRTKEYFMT

Print key format. Changes made to this system value take effect for jobs started after the change is made.

- *PRTHDR means that header information is printed when the print key is pressed.
- *PRTBDR means that border information is printed when the print key is pressed.
- *PRTALL means that border information and header information are printed when the print key is pressed.
- *NONE means that border information and header information are not printed when the print key is pressed.

QPRTTXT

Up to 30 characters of text that can be printed at the bottom of listings and separator pages. Changes made to this system value take effect for jobs started after the change is made.

QPWDEXPITV

The number of days for which a password is valid. Changes made to this system value take effect immediately.

- *NOMAX means a password can be used an unlimited number of days.
- 1-366 means the number of days before the password ends.

QPWDLMTAJC

Limits the use of adjacent numbers in a password. Changes made to this system value take effect the next time a password is changed.
- 0 means adjacent numbers are allowed.
- 1 means adjacent numbers are not allowed.

QPWDLMTCHR

Limits the use of certain characters in a password. Changes made to this system value take effect the next time a password is changed.
- *NONE means there are no restricted characters.
- restricted-characters means up to 10 restricted characters enclosed in apostrophes can be specified. Valid characters are: A-Z, 0-9, and special characters #, $, @, or underscore (_).

Note: This system value is ignored if the system is operating at QPWDLVL 2 or 3.

QPWDLMTREP

Limits the use of repeating characters in a password. Changes made to this system value take effect the next time a password is changed.
- 0 means characters can be used more than once.
- 1 means characters cannot be used more than once.

QPWDLVL

Specifies the password level.

Changing this system value requires careful consideration. If your system connects to other systems in a network then all systems must be able to run with the password rules that will be in effect.

See the iSeries Security Reference publication for additional considerations prior to changing this system value.

Changes to this system value will take effect on the next IPL.
- 0 means passwords from 1-10 characters are allowed.
- 1 means passwords from 1-10 characters are allowed. iSeries NetServer passwords for Windows 95/98/ME clients will be removed from the system making the product unavailable for use.
- 2 means passwords from 1-128 characters are allowed. Passwords can consist of any character and will be case sensitive.
- 3 means passwords from 1-128 characters are allowed. Passwords can consist of any character and will be case sensitive. iSeries NetServer passwords for Windows 95/98/ME clients will be removed from the system making the product unavailable for use.

QPWDMAXLEN

The maximum number of characters in a password. Changes made to this system value take effect the next time a password is changed.
- 1-128 means a value from 1 to 128 can be specified as the maximum number of characters in a password.

  If the system is operating at QPWDLVL 0 or 1, the valid range is 1-10. If the system is operating at QPWDLVL 2 or 3, the valid range is 1-128.

QPWDMINLEN

The minimum number of characters in a password. Changes made to this system value take effect the next time a password is changed.

- 1-128 means a value from 1 to 128 can be specified as the minimum number of characters in a password.

  If the system is operating at QPWDLVL 0 or 1, the valid range is 1-10. If the system is operating at QPWDLVL 2 or 3, the valid range is 1-128.

QPWDPOSDIF

Controls the position of characters in a new password. Changes made to this system value take effect the next time a password is changed.

- 0 means the same characters can be used in a position corresponding to the same position in the previous password.
- 1 means the same character cannot be used in a position corresponding to the same position in the previous password.

QPWDRQDDGT

Require number in a new password. Changes made to this system value take effect the next time a password is changed.

- 0 means numbers are not required.
- 1 means one or more numbers are required.

QPWDRQDDIF

Controls whether the password must be different than the previous passwords. Changes made to this system value take effect the next time a password is changed.

- 0 means a password can be the same as one previously used.
- 1 means a password must be different than the previous 32 passwords.
- 2 means a password must be different than the previous 24 passwords.
- 3 means a password must be different than the previous 18 passwords.
- 4 means a password must be different than the previous 12 passwords.
- 5 means a password must be different than the previous 10 passwords.
- 6 means a password must be different than the previous 8 passwords.
- 7 means a password must be different than the previous 6 passwords.
- 8 means a password must be different than the previous 4 passwords.

QPWDVLDPGM

Password validation program provides the ability for a user-written program to do additional validation on passwords. Changes made to this system value take effect the next time a password is changed. See Password validation program for additional information.

QPWRDWNLMT

Maximum amount of time (in seconds) allowed for PWRDWNSYS *IMMED. This is the time used to wait for power down to complete normally after either of the following happens:

- A Power Down System (PWRDWNSYS) command with *IMMED specified for the **How to end** (OPTION) parameter is entered.
- A PWRDWNSYS command with *CNTRLD specified for the **How to end** (OPTION) parameter is entered and the time specified for the **Controlled end delay time** (DELAY) parameter has ended.

Changes to this value take effect when a PWRDWNSYS command is entered.

QPWRRSTIPL

Automatic initial program load (IPL) after power restored allowed. Changes made to this system value take effect the next time there is a power failure.

- 0 means no auto-IPL after power restored.
- 1 means auto-IPL after power restored.

On partitioned AS/400 7xx and iSeries 8xx servers, this system value can only be changed from the primary partition. Whether or not a secondary partition is IPLed at the same time as the primary partition depends on the secondary partition's configuration value for IPL action.

On partitioned eServer i5 servers, this system value must be changed from the service processor's Advanced System Management (ASM) interface.

For more information on partitions, see the Logical Partitions topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

QQRYDEGREE

Query parallel processing degree. The value specifies the parallel processing degree available to users of the system.

- *NONE means no parallel processing is allowed for database query processing or database file keyed access path builds or rebuilds.
- *IO means any number of tasks can be used when the database query optimizer chooses to use I/O parallel processing for queries. SMP parallel processing is not allowed, including when building or rebuilding database file keyed access paths.
- *OPTIMIZE means the query optimizer can choose to use any number of tasks for either I/O or SMP parallel processing to process the query or database file keyed access path build or rebuild. Use of parallel processing and the number of tasks used is determined with respect to the number of processors available in the pool in which the job is run, and whether the expected elapsed time for the query or database file keyed access path build or rebuild, is limited by CPU processing or I/O resources.
- *MAX means the query optimizer can choose to use either I/O or SMP parallel processing to process the query. The choices made by the query optimizer will be similar to those made for the value *OPTIMIZE except the optimizer will assume that all active memory in the pool can be used to process the query or database file keyed access path build or rebuild.

QQRYTIMLMT

Query processing time limit.

- *NOMAX means the maximum query interval is used.
- 0-2147352578 means the number of seconds allowed for query processing.

QRCLSPLSTG

Automatic deletion of empty spooled members is allowed based on the member retention interval. Changes made to this system value take effect immediately.

- *NONE means no retention interval.

  Note: Using this value can have adverse effects on system performance. More information is in the Files and file systems topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.
- *NOMAX means all empty members are kept.
- 1-366 means the number of days that empty spooled members are kept for new spooled file use.

QRETSVRSEC

Retain server security data indicator. This value determines whether the security data needed by a server to authenticate a user on a target system through client/server interfaces can be retained on this system.

- 0 means that the server security data is not retained.
- 1 means that the server security data is retained.

QRMTSRVATR

Remote service attribute. The QRMTSRVATR system value controls the remote service problem analysis ability. The value allows the system to be analyzed remotely.

- 0 means the remote service attribute is off.

- 1 means the remote service attribute is on.

QRMTIPL

Remote power on and IPL indicator. Changes made to this system value take effect immediately.

- 0 means remote power on and IPL are not allowed.
- 1 means remote power on and IPL are allowed.

  Note: Any telephone call will cause the system to IPL.

On partitioned AS/400 7xx and iSeries 8xx servers, this system value can only be changed from the primary partition. Whether or not a secondary partition is IPLed at the same time as the primary partition depends on the secondary partition's configuration value for IPL action.

On partitioned eServer i5 servers, this system value must be changed from the service processor's Advanced System Management (ASM) interface.

For more information on partitions, see the Logical Partitions topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

QRMTSIGN

Remote sign-on control. Changes made to this system value take effect immediately.

- *FRCSIGNON means normal sign-on required.
- *SAMEPRF means when the source and target user profile are the same, the sign-on can be bypassed for remote sign-on attempts.
- *REJECT means no remote sign-on is allowed.
- *VERIFY means after verifying that the user has access to the system, the system allows the user to bypass the sign-on.
- program means you can specify a program to decide which remote sessions will be allowed and which user profiles can be automatically signed-on from which locations.

QSAVACCPTH

Save access paths. Changes made to this system value take effect at the start of the next save operation.

- 0 means do not save logical file access paths that are dependent on the physical files that are being saved.
- 1 means save logical file access paths that are dependent on the physical files that are being saved.

QSCANFS

Scan file systems. This system value specifies the integrated file systems in which objects will be scanned when exit programs are registered with any of the integrated file system scan-related exit points. Changes made to this system value take effect immediately. See Scan file systems for additional information.

QSCANFSCTL

Scan file systems control. This system value controls the integrated file system scanning on the system when exit programs are registered with any of the integrated file system scan-related exit points. These controls apply to integrated file system objects in the file systems covered by the QSCANFS(Scan file systems) system value. Changes made to this system value take effect immediately. See Scan file systems control for additional information.

QSCPFCONS

IPL action with console problem. Changes to this system value take effect before the next IPL.

- 0 means end system.
- 1 means continue the unattended IPL.

QSECOND

Second of the minute. Changes made to this system value take effect immediately.

QSECURITY

System security level. Changes made to this system value take effect at the next IPL.

- 20 means the system requires a password to sign-on.
- 30 means password security at sign-on and object security at each access. You must have authority to access all system resources.
- 40 means password security at sign-on and object security at each access. Programs that try to access objects through interfaces that are not supported will fail.
- 50 means the system requires a password to sign on and users must have authority to access objects and system resources. The security and integrity of the QTEMP library and user domain objects are enforced. Programs that try to access objects through interfaces that are not supported or that try to pass unsupported parameter values to supported interfaces will fail.

QSFWERRLOG

Software error log. Indicates whether system-detected software problems are entered in the error log. Changes made to this system value take effect immediately.

- *LOG means that when a software error is detected by the system, the error is evaluated to determine if it should be logged unconditionally, or if the decision to log the error should be deferred to the policy based Service Monitor.

  If the error is to be logged unconditionally, a PARable message is sent to QSYSOPR and an entry is created in the problem log. If the reporting component provides error data, a spooled file is created to contain the data. The spooled file name is stored in the error log and problem log entries.

  If the error is to be conditionally logged, the decision to log the error will be made by the policy based Service Monitor. If the decision is to log the problem, an entry is created in the problem log. The problem data will be stored in a problem data library and the problem record entry will be updated with the name of the library.

- *NOLOG means no logging will occur if a software error is detected.

QSHRMEMCTL

Shared memory control. Specifies whether or not users can use shared memory, or use mapped memory that has write capability. Changes made to this system value take effect immediately.

- 0 means that users cannot use shared memory, or use mapped memory that has write capability.
- 1 means that users can use shared memory or mapped memory that has write capability.

QSPCENV

Special environment. The system environment used as the default for all users. Changes made to this system value take effect the next time a user signs on to the system.

- *NONE means no special environment is entered when you sign on.
- *S36 means the System/36 environment is entered when you sign on.

QSPLFACN

Spooled file action. Specifies whether spooled files are kept with a job or detached from the job. Keeping spooled files with jobs allows job commands such as the Work with Submitted Jobs (WRKSBMJOB) command to work with the spooled files even after the job has ended. Detaching spooled files from jobs reduces the use of system resources by allowing job structures to be recycled when the job ends. A change to this system value takes effect for all jobs that become active after the change. The shipped value is *KEEP.

- *KEEP means that when the job ends, as long as at least one spooled file for the job exists in the system auxiliary storage pool (ASP number 1) or in a basic user ASP (ASP numbers 2-32), the spooled files are kept with the job and the status of the job is updated to indicate that the job has completed. If all remaining spooled files for the job are in independent ASPs (ASP numbers 33-255), the spooled files will be detached from the job and the job will be removed from the system.

- *DETACH means the spooled files are detached from the job when the job ends.

QSRLNBR

System serial number. This value cannot be changed. If is retrieved from the data fields by the system when installing the operating system licensed program. You can display QSRLNBR, or you can retrieve this value in user-written programs. The system serial number is the same in each partition on a system.

QSRTSEQ

Sort sequence. This system value specifies the default sort sequence algorithm to be used by the system. Changes made to this system value take effect for jobs started after the change is made.

QSRVDMP

Service dumps. Indicates whether service dumps for escape messages that are not monitored are created. Changes made to this system value take effect immediately.

- *DMPUSRJOB means that service dumps are created only for user jobs, not system jobs.
- *DMPSYSJOB means that service dumps are created only for system jobs, not user jobs. System jobs include the operating system, subsystem monitors, LU service process, spooled readers and writers, and the SCPF job.
- *DMPALLJOB means that service dumps are created for all jobs.
- *NONE means no service dumps are created.

QSTGLOWACN

Auxiliary storage lower limit action. Specifies the action to take when the available storage in the system ASP goes below the auxiliary storage lower limit. A change to this system value takes effect immediately. The shipped value is *MSG.

- *MSG: Send message CPI099C to QSYSMSG and QSYSOPR message queue. This message is also sent for the other actions.
- *CRITMSG: Send critical message CPI099B to the user specified in the service attribute to receive critical messages.
- *REGFAC: Submit a job to call exit programs registered for the QIBM_QWC_QSTGLOWACN exit point.
- *ENDSYS: End the system to the restricted state.
- *PWRDWNSYS: Power down the system immediately and restart it.

QSTGLOWLMT

Auxiliary storage lower limit. Specifies the percent of available storage remaining in the system ASP when the auxiliary storage lower limit action is taken. A change to this system value takes effect immediately. The shipped value is 5.0.

- Lower limit: Percentage of available storage remaining in the system ASP when the action specified in QSTGLOWACN is taken. The percent of storage currently used in the system ASP can be viewed with the Work with System Status (WRKSYSSTS) command.

QSTRPRTWTR

Start print writers at initial program load (IPL). This system value is set by the system at the time of IPL or is set by the user on the IPL Options display. This system value cannot be changed using the Change System Value (CHGSYSVAL) command.

- 0 means print writers were not started.
- 1 means print writers were started.

QSTRUPPGM

Start-up program name from autostart job in the controlling subsystem. Both an object name and library name can be specified. Changes made to this system value take effect at the next IPL.

QSTSMSG

Indicates whether status messages are shown. Changes made to this system value take effect the next time a user signs on to the system.

- *NORMAL means status messages will be shown.
- *NONE means status messages will not be shown.

QSVRAUTITV

Server authentication interval. The operating system no longer uses this system value. Changes made to this system value have no effect.

QSYSLIBL

System part of the library list. Changes made to this system value take effect for jobs started after the change is made.

QTHDRSCADJ

Thread resources adjustment. This system value specifies whether or not the system should dynamically make adjustments to the affinity or preference of threads currently running in the system to certain processors and memory. If some resources are being utilized more than others, the system may reassign some of the threads running on the more heavily utilized resources to have affinity to the less utilized resources. Changes made to this system value take effect immediately. The shipped value is '1.'

- '0' means no automatic adjustment of threads is made by the system. Threads will continue to have affinity to the resources which they are currently assigned to until they end or until the system value is changed.
- '1' means the system dynamically makes adjustments of threads' affinity to the system's resources. It does not change the grouping or level of affinity in the threads.

QTHDRSCAFN

Thread resources affinity. The affinity or preference of threads to certain processors and memory. Changes made to this system value take effect immediately for threads in jobs that are started after the change, but has no effect on threads currently running.

- *NOGROUP - Secondary threads will not necessarily have affinity to the same group of processors and memory as their initiating thread.
- *GROUP - Secondary threads will have affinity to the same group of processors and memory as their initiating thread.

The thread resources affinity level can be set to the following values:

- *NORMAL - A thread will use any processor or memory if the resources it has affinity to are not readily available.
- *HIGH - A thread will only use the resources it has affinity to, and will wait until they become available if necessary.

QTIMADJ

Time adjustment. This system value can be used to identify software that adjusts the system clock to keep it synchronized with an external time source. This value should be maintained by time adjustment software and is intended as an aid to prevent having multiple time adjustment applications conflict with each other. There are no checks performed by the system to verify this value or that software is or is not performing time adjustments. IBM time adjustment offerings will use identifiers that start with QIBM such as 'QIBM_OS400_SNTP'. Other software suppliers should follow a similar naming convention of company name and product name.

Time adjustment software should check QTIMADJ prior to starting. If QTIMADJ has an identifier for other time adjustment software, then the software being started should notify the user of this potential conflict and confirm that this time adjustment software should be started. When QTIMADJ is *NONE the software should update QTIMADJ to identify that it is now responsible for adjusting the system clock. Time adjustment software should check QTIMADJ again prior to ending. QTIMADJ should be set to *NONE only if the current value identifies this time adjustment software that is ending. Changes made to this system value take effect immediately. The shipped value is *NONE.

- *NONE - Indicates that time adjustment software has not been identified.

- Identifier - Identify the software that will be used to adjust the system clock.

QTIME

Time of day. Changes made to this system value take effect immediately.

QTIMSEP

Time separator. Changes made to this system value take effect for jobs started after the change is made.

This value affects jobs for which *SYSVAL is specified as the time separator. When specifying time on commands, users must use the time separator specified for their job or no time separator. If a time separator different from the job's time separator is used to specify time on a command, the command will fail.

QTIMZON

Time zone. This specifies the name of the time zone description used to calculate local system time. A change to a different time zone description may result in a different offset that is associated with this new time zone description. The system value QUTCOFFSET will be changed as well to match this new offset. Changes made to this system value take effect immediately.

QTOTJOB

The total number of jobs for which storage must be allocated. Changes made to this system value take effect at the next IPL.

QTSEPOOL

Indicates whether interactive jobs should be moved to another main storage pool when they reach time slice end. Changes made to this system value take effect for jobs started after the change is made.

- *NONE means jobs are not moved when time slice end is reached.
- *BASE means jobs are moved when time slice end is reached.

QUPSDLYTIM

Uninterruptible power supply delay time. Changes made to this system value take effect the next time there is a power failure.

- *BASIC powers only the PRC, IOP cards, and Load Source Disk.
- *CALC means the appropriate wait time will be calculated.
- *NOMAX means the system will not start any action on its own.
- 0 means the system will power down automatically when system utility power fails.
- 1-99999 means specify the delay time in seconds before the system powers down.

On some partitioned systems, this system value can only be changed from the primary partition.

For more information on partitions, see the Logical Partitions topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

QUPSMSGQ

Message queue for uninterruptible power supply messages. Changes made to this system value take effect the next time there is a power failure.

QUSEADPAUT

Defines which users can create, change and update programs and service programs with the (use adopted authority) USEADPAUT(*YES) attribute. When a program or service program has a use adopted authority attribute of *YES, the program/service program can use any adopted authority that is being passed to it from a program/ service program higher in the call stack.

This system value has no effect on the following:

- Existing programs/service programs created with the USEADPAUT(*YES) attribute. Users are responsible for deciding which existing programs/service programs should be changed to have USEADPAUT(*NO).

- Restoring a program/service program that uses adopted authority. These program/service programs can still be restored on your system.
- Duplicating a program/service program that uses adopted authority. The USEADPAUT attribute of the existing program/service program is copied to the new object.

The following values can be specified:

- *NONE means there is no restriction on who can create, change or update a program/service program to use adopted authority. Any user can create, change or update a program/service program to have the USEADPTAUT(*YES) attribute.
- Name means you can specify the name of the authorization list which will control which users can set the USEADPAUT(*YES) attribute. The user needs *USE authority to the authorization list to be able to create, change or update programs/service programs with the USEADPAUT(*YES) attribute. Authority to the authorization list cannot come from adopted authority. That is, if you are running a program that adopts authority, the adopted authority is not used when checking authority to the authorization list.

QUSRLIBL

User part of the library list. Changes made to this system value take effect for jobs started after the change is made.

QUTCOFFSET

Indicates the number of hours (in 24-hour format) and minutes that the current system time is offset from the Coordinated Universal Time (UTC).

- +hhmm means that the current system time is hh hours and mm minutes ahead of UTC.
- -hhmm means that the current system time is hh hours and mm minutes behind UTC.

Note: This system value must be the same as the offset that is associated with the time zone description specified in the system value QTIMZON. A change to a different time zone description for QTIMZON may result in a different associated offset. The system value QUTCOFFSET will be changed as well to match this new offset. QUTCOFFSET cannot be changed to a value that is different than the offset currently associated with QTIMZON. If an attempt is made to do so, the diagnostic message CPD1687 will be issued.

QVFYOBJRST

Verify object on restore. This system value specifies the policy to be used for object signature verification during a restore operation. This value applies to objects of types: *CMD, *PGM, *SRVPGM, *SQLPKG and *MODULE. It also applies to *STMF objects which contain Java programs. This value also specifies the policy for PTFs applied to the system including Licensed Internal Code fixes. Changes made to this system value take effect immediately. See Verify object on restore for additional information.

QYEAR

Year. Changes made to this system value take effect immediately.

CL variable for returned value (RTNVAR)

Specifies the name of the CL program variable that receives the value of the system value being returned. The type and length for the CL variable when it was declared must be compatible with that of the system value being received. The attributes of individual system values are described in the Work Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

In general, the return variable type must match the system value type. For character system values that are 1 character long, the CL variable can be a character or logical variable. For character and logical system values, the length of the CL variable must equal the length of the system value. For decimal system values, the CL variable length must be greater than or equal to the length of the system value.

The following table lists the type and length of each system value.

```
SYSTEM VALUE    TYPE        LENGTH
----------------------------------------------
QABNORMSW       Character   1
QACGLVL         Character   80
QACTJOB         Decimal     (5 0)
QADLACTJ        Decimal     (5 0)
QADLSPLA        Decimal     (5 0)
QADLTOTJ        Decimal     (5 0)
QALWJOBITP      Character   1
QALWOBJRST      Character   150
QALWUSRDMN      Character   500
QASTLVL         Character   10
QATNPGM         Character   20
QAUDCTL         Character   50
QAUDENDACN      Character   10
QAUDFRCLVL      Decimal     (5 0)
QAUDLVL         Character   160
QAUDLVL2        Character   990
QAUTOCFG        Character   1
QAUTORMT        Character   1
QAUTOSPRPT      Character   1
QAUTOVRT        Decimal     (5 0)

QBASACTLVL      Decimal     (5 0)
QBASPOOL        Decimal     (10 0)
QBOOKPATH       Character   (315)

QCCSID          Decimal     (5 0)
QCENTURY        Character   1
QCFGMSGQ        Character   20
QCHRID          Character   20
QCHRIDCTL       Character   10
QCMNARB         Character   10
QCMNRCYLMT      Character   20
QCNTRYID        Character   2
QCONSOLE        Character   10
QCRTAUT         Character   10
QCRTOBJAUD      Character   10
QCTLSBSD        Character   20
QCURSYM         Character   1

QDATE           Character   5 (for Julian dates) or 6
QDATETIME       Character   20
QDATFMT         Character   3
QDATSEP         Character   1
QDAY            Character   2 or 3 (for Julian dates)
QDAYOFWEEK      Character   4
QDBFSTCCOL      Character   10
QDBRCVYWT       Character   1
QDECFMT         Character   1
QDEVNAMING      Character   10
QDEVRCYACN      Character   20
QDSCJOBITV      Character   10
QDSPSGNINF      Character   1
QDYNPTYADJ      Character   1
QDYNPTYSCD      Character   1

QENDJOBLMT      Decimal     (5 0)

QFRCCVNRST      Character   1

QHOUR           Character   2
QHSTLOGSIZ      Decimal     (8 0)

QIGC            Character   1
QIGCCDEFNT      Character   20
QIGCFNTSIZ      Decimal     (4 1)
QINACTITV       Character   10
```

```
QINACTMSGQ      Character   20
QIPLDATTIM      Character   20
QIPLSTS         Character   1
QIPLTYPE        Character   1

QJOBMSGQFL      Character   10
QJOBMSGQMX      Decimal     (5 0)
QJOBMSGQSZ      Decimal     (5 0)
QJOBMSGQTL      Decimal     (5 0)
QJOBSPLA        Decimal     (5 0)

QKBDBUF         Character   10
QKBDTYPE        Character   3

QLANGID         Character   3
QLEAPADJ        Decimal     (5 0)
QLIBLCKLVL      Character   1
QLMTDEVSSN      Character   1
QLMTSECOFR      Character   1
QLOCALE         Character   1024
QLOGOUTPUT      Character   10

QMAXACTLVL      Decimal     (5 0)
QMAXJOB         Decimal     (10 0)
QMAXSGNACN      Character   1
QMAXSIGN        Character   6
QMAXSPLF        Decimal     (10 0)
QMCHPOOL        Decimal     (10 0)
QMINUTE         Character   2
QMLTTHDACN      Character   1
QMODEL          Character   4
QMONTH          Character   2

QPASTHRSVR      Character   10
QPFRADJ         Character   1
QPRBFTR         Character   20
QPRBHLDITV      Decimal     (5 0)
QPRCFEAT        Character   4
QPRCMLTTSK      Character   1
QPRTDEV         Character   10
QPRTKEYFMT      Character   10
QPRTTXT         Character   30
QPWDEXPITV      Character   6
QPWDLMTAJC      Character   1
QPWDLMTCHR      Character   10
QPWDLMTREP      Character   1
QPWDLVL         Decimal     (5 0)
QPWDMAXLEN      Decimal     (5 0)
QPWDMINLEN      Decimal     (5 0)
QPWDPOSDIF      Character   1
QPWDRQDDGT      Character   1
QPWDRQDDIF      Character   1
QPWDVLDPGM      Character   20
QPWRDWNLMT      Decimal     (5 0)
QPWRRSTIPL      Character   1

QQRYDEGREE      Character   10
QQRYTIMLMT      Character   10

QRCLSPLSTG      Character   10
QRETSVRSEC      Character   1
QRMTIPL         Character   1
QRMTSRVATR      Character   1
QRMTSIGN        Character   20

QSAVACCPTH      Character   1
QSCANFS         Character   200
QSCANFSCTL      Character   200
QSCPFCONS       Character   1
QSECOND         Character   2
QSECURITY       Character   2
QSETJOBATR      Character   160
```

```
QSFWERRLOG    Character    10
QSHRMEMCTL    Character    1
QSPCENV       Character    10
QSPLFACN      Character    10
QSRLNBR       Character    8
QSRTSEQ       Character    20
QSRVDMP       Character    10
QSTGLOWACN    Character    10
QSTGLOWLMT    Decimal      (7 4)
QSTRPRTWTR    Character    1
QSTRUPPGM     Character    20
QSTSMSG       Character    10
QSVRAUTITV    Decimal      (6 0)
QSYSLIBL      Character    150

QTHDRSCADJ    Character    1
QTHDRSCAFN    Character    20
QTIMADJ       Character    30
QTIME         Character    6, 7, 8, or 9 (for seconds,
                           tenths, hundredths, or
                           thousandths of a second)
QTIMSEP       Character    1
QTIMZON       Character    10
QTOTJOB       Decimal      (5 0)
QTSEPOOL      Character    10

QUPSDLYTIM    Character    20
QUPSMSGQ      Character    20
QUSEADPAUT    Character    10
QUSRLIBL      Character    250
QUTCOFFSET    Character    5

QVFYOBJRST    Character    1

QYEAR         Character    2
```

This is a required parameter.

Examples

```
RTVSYSVAL   SYSVAL(QDATE)   RTNVAR(&DATE)
```

This command retrieves the date value from the system value QDATE and copies it into the CL variable &DATE. The CL variable must be declared as a 6-character variable to match the attributes of the QDATE system value.

Error messages

*ESCAPE Messages

CPF1028

&1 not valid for parameter SYSVAL.

CPF1074

SYSVAL(QMONTH) not valid for Julian date format.

CPF1094

CL variable not same type as system value &1.

CPF1095

CL variable length not valid for system value &1.

CPF1842

> Cannot access system value &1.

CPF268D

> Unable to access system value &1.

Retrieve Table Source (RTVTBLSRC)

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

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. Additional, this command will convert the source of a *SRTSEQ table to the format used to create a *UCSSRTSEQ table.

Parameters

Keyword	Description	Choices	Notes
TBL	Table	*Qualified object name*	Required, Positional 1
	Qualifier 1: Table	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCFILE	Source file	*Qualified object name*	Required, Positional 2
	Qualifier 1: Source file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***TBL**	Optional, Positional 3
CVTTOUCS	Convert to UCS-2	*YES, ***NO**	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Table (TBL)

Specifies the qualified name of the table whose source is being retrieved.

Source file (SRCFILE)

Specifies the qualified name of the previously created database source file into which the table source is being written.

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

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

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

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

Source member (SRCMBR)

Specifies the name of the database source file member into which the table source is being written. If not specified, the table name is assumed. If the member existed before running the command, it is cleared before any source statements are written into it. If the member did not exist, it is created.

The possible values are:

*TBL   The name of the table is used as the member name.

source-member-name
        Specify the name of the source file member to contain the retrieved table source.

Convert to UCS-2 (CVTTOUCS)

Specifies whether the source of a *SRTSEQ table should be converted when placed in the file from the format of a *SRTSEQ table to that of a *UCSSRTSEQ. You would want to do this if you want to create a *UCSSRTSEQ but want to base it on an existing *SRTSEQ table.

The possible values are:

*NO   The information is not converted when placed in the file.

*YES   The information is converted when placed in the file.

Text 'description' (TEXT)

Specifies the text that briefly describes the object.

*BLANK
        No text is specified.

'description'
        Specify no more than 50 characters of text, enclosed in apostrophes.

Examples

```
RTVTBLSRC   TBL(QSYS/TABLE1)   SRCFILE(TBLSRC)   SRCMBR(TABLEOUT)
            CVTTOUCS(*NO)
```

This command retrieves the source statements from the table named TABLE1 in library QSYS. The retrieved source is placed into the file named TBLSRC and is named as member TABLEOUT and not converted to a *UCSSRTSEQ format.

Error messages

Unknown

Retrieve User Profile (RTVUSRPRF)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve User Profile (RTVUSRPRF) command is used in a control language (CL) program or a REXX procedure to get one or more of the values that are stored and associated with a user. The values are returned in the specified variables for the desired user.

The parameter prompt text lists the minimum length for the variables next to the appropriate parameters you want to retrieve. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

- Read (*READ) authority is required for the user profile specified on the USRPRF parameter.

Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	*Name*, ***CURRENT**	Optional, Positional 1
RTNUSRPRF	CL var for RTNUSRPRF (10)	*Character value*	Optional
SPCAUT	CL var for SPCAUT (100)	*Character value*	Optional
MAXSTG	CL var for MAXSTG (11 0)	*Decimal number*	Optional
STGUSED	CL var for STGUSED (15 0)	*Decimal number*	Optional
PTYLMT	CL var for PTYLMT (1)	*Character value*	Optional
INLPGM	CL var for INLPGM (10)	*Character value*	Optional
INLPGMLIB	CL var for INLPGMLIB (10)	*Character value*	Optional
JOBD	CL var for JOBD (10)	*Character value*	Optional
JOBDLIB	CL var for JOBDLIB (10)	*Character value*	Optional
GRPPRF	CL var for GRPPRF (10)	*Character value*	Optional
OWNER	CL var for OWNER (10)	*Character value*	Optional
GRPAUT	CL var for GRPAUT (10)	*Character value*	Optional
ACGCDE	CL var for ACGCDE (15)	*Character value*	Optional
MSGQ	CL var for MSGQ (10)	*Character value*	Optional
MSGQLIB	CL var for MSGQLIB (10)	*Character value*	Optional
OUTQ	CL var for OUTQ (10)	*Character value*	Optional
OUTQLIB	CL var for OUTQLIB (10)	*Character value*	Optional
TEXT	CL var for TEXT (50)	*Character value*	Optional
PWDCHGDAT	CL var for PWDCHGDAT (6)	*Character value*	Optional
USRCLS	CL var for USRCLS (10)	*Character value*	Optional
ASTLVL	CL var for ASTLVL (10)	*Character value*	Optional
SPCENV	CL var for SPCENV (10)	*Character value*	Optional

Keyword	Description	Choices	Notes
CURLIB	CL var for CURLIB (10)	*Character value*	Optional
INLMNU	CL var for INLMNU (10)	*Character value*	Optional
INLMNULIB	CL var for INLMNULIB (10)	*Character value*	Optional
LMTCPB	CL var for LMTCPB (10)	*Character value*	Optional
DLVRY	CL var for DLVRY (10)	*Character value*	Optional
SEV	CL var for SEV (2 0)	*Decimal number*	Optional
PRTDEV	CL var for PRTDEV (10)	*Character value*	Optional
ATNPGM	CL var for ATNPGM (10)	*Character value*	Optional
ATNPGMLIB	CL var for ATNPGMLIB (10)	*Character value*	Optional
USROPT	CL var for USROPT (240)	*Character value*	Optional
DSPSGNINF	CL var for DSPSGNINF (7)	*Character value*	Optional
PWDEXPITV	CL var for PWDEXPITV (5 0)	*Decimal number*	Optional
PWDEXP	CL var for PWDEXP (4)	*Character value*	Optional
STATUS	CL var for STATUS (10)	*Character value*	Optional
PRVSIGN	CL var for PRVSIGN (13)	*Character value*	Optional
NOTVLDSIGN	CL var for NOTVLDSIGN (11 0)	*Decimal number*	Optional
LMTDEVSSN	CL var for LMTDEVSSN (7)	*Character value*	Optional
KBDBUF	CL var for KBDBUF (10)	*Character value*	Optional
LANGID	CL var for LANGID (10)	*Character value*	Optional
CNTRYID	CL var for CNTRYID (10)	*Character value*	Optional
CCSID	CL var for CCSID (5 0)	*Decimal number*	Optional
SRTSEQ	CL var for SRTSEQ (10)	*Character value*	Optional
SRTSEQLIB	CL var for SRTSEQLIB (10)	*Character value*	Optional
OBJAUD	CL var for OBJAUD (10)	*Character value*	Optional
AUDLVL	CL var for AUDLVL (640)	*Character value*	Optional
GRPAUTTYP	CL var for GRPAUTTYP (10)	*Character value*	Optional
SUPGRPPRF	CL var for SUPGRPPRF (150)	*Character value*	Optional
UID	CL var for UID (10 0)	*Decimal number*	Optional
GID	CL var for GID (10 0)	*Decimal number*	Optional
SETJOBATR	CL var for SETJOBATR (160)	*Character value*	Optional
CHRIDCTL	CL var for CHRIDCTL (10)	*Character value*	Optional
LCLPWDMGT	CL var for LCLPWDMGT (10)	*Character value*	Optional

User profile (USRPRF)

Specifies the user profile whose information you want to retrieve. If a variable is specified, it must be 10 characters in length and contain a user name or the value *CURRENT.

*CURRENT
>	The user profile that is currently running is used.

name	Specify the name of the user profile whose information you want to retrieve.

CL var for RTNUSRPRF (10) (RTNUSRPRF)

Specifies the name of a variable that is used to retrieve the name of the user profile. In CL programs, this should be a 10-character variable. If *CURRENT is specified for the **User profile (USRPRF)** parameter, the value returned is the currently running user profile name. If a name is specified, that name is returned for this parameter.

CL var for SPCAUT (100) (SPCAUT)

Specifies the name of a variable that is used to retrieve the list of special authorities the user has. In CL programs, this should be a 100-character variable. The format returned is a list of up to 10 entries. Each entry is 10 characters in length and each entry contains a special authority. If there are fewer than 10 special authorities in the list, the remaining entries are padded on the right with blanks. If the user has no special authorities, the first entry contains the value of *NONE, followed by blanks.

CL var for MAXSTG (11 0) (MAXSTG)

Specifies the name of a variable that is used to retrieve the maximum amount of auxiliary storage that can be assigned to store permanent objects owned by the specified user. In CL programs, this should be a decimal variable length of (11 0). The value returned is either an 11-digit value or a value of -1 if it is *NOMAX. The variable must be an 11-digit value with no decimal positions.

CL var for STGUSED (15 0) (STGUSED)

Specifies the name of a variable that is used to get the amount of auxiliary storage that is currently being used to store permanent objects owned by the specified user profile. In CL programs, this should be a decimal variable length of (15 0). The value is returned in kilobytes (1 kilobyte (KB) equals 1024 bytes).

CL var for PTYLMT (1) (PTYLMT)

Specifies the name of a variable used to retrieve the highest scheduling priority the user is allowed to have for each job submitted to the system. In CL programs, the variable has a length of 1 character. This value controls the job processing priority that any job running under this user can have. This means that values specified in the JOBPTY and OUTPTY parameters of any job command cannot exceed the PTYLMT value specified for the user under which the job is run. The scheduling priority can have a value ranging from 0 through 9, where 0 is the highest priority and 9 is the lowest priority.

CL var for INLPGM (10) (INLPGM)

Specifies the name of a variable that is used to retrieve the name of the initial program that starts when the specified user signs on to the system. In CL programs, this should be a 10-character variable. If no initial program name is associated with the specified user, the value returned in the variable is *NONE.

CL var for INLPGMLIB (10) (INLPGMLIB)

Specifies the name of a variable that is used to retrieve the name of the library that contains the initial program associated with the specified user. In CL programs, this should be a 10-character variable. If there is no initial program associated with the specified user, blanks are returned in the variable.

CL var for JOBD (10) (JOBD)

The name of a variable that is used to retrieve the name of the job description associated with the specified user. In CL programs, this should be a 10-character variable.

CL var for JOBDLIB (10) (JOBDLIB)

Specifies the name of a variable that is used to retrieve the name of the library that contains the job description associated with the specified user. In CL programs, this should be a 10-character variable.

CL var for GRPPRF (10) (GRPPRF)

Specifies the name of a variable that is used to retrieve the name of the group profile. In CL programs, this should be a 10-character variable. If no group profile exists for the specified user profile, a value of *NONE is returned in the variable.

CL var for OWNER (10) (OWNER)

Specifies the name of a variable that is used to retrieve the special value of *USRPRF or *GRPPRF. This parameter The owner of newly created objects. This is either the specified user or the user's group profile. In CL programs, this should be a 10-character variable. If no group profile exists for the specified user profile, the value returned in the variable is *USRPRF.

CL var for GRPAUT (10) (GRPAUT)

Specifies the name of a variable used to retrieve the authority granted to the group profile for newly-created objects. The special value of *NONE, *CHANGE, *ALL, *USE, or *EXCLUDE is returned in the variable. If there is no group profile for the specified user, or if the group profile is the owner of the objects the specified user creates, the special value returned is *NONE. In CL programs, this should be a 10-character variable.

Top

CL var for ACGCDE (15) (ACGCDE)

The name of a variable that is used to retrieve the value of the accounting code assigned to the specified user. In CL programs, this should be a 15-character variable. If no accounting code exists for the user profile, blanks are returned.

Top

CL var for MSGQ (10) (MSGQ)

The name of a variable that is used to retrieve the name of the message queue associated with the specified user. In CL programs, this should be a 10-character variable.

Top

CL var for MSGQLIB (10) (MSGQLIB)

Specifies the name of a variable that is used to retrieve the name of the library that contains the message queue associated with the specified user. In CL programs, this should be a 10-character variable.

Top

CL var for OUTQ (10) (OUTQ)

Specifies the name of a variable that is used to retrieve the name of the output queue associated with the specified user. In CL programs, this should be a 10-character variable. The special value *DEV or *WRKSTN is returned in the variable.

Top

CL var for OUTQLIB (10) (OUTQLIB)

Specifies the name of a variable that is used to retrieve the name of the library that contains the output queue associated with the specified user. In CL programs, this should be a 10-character variable. Blanks are returned if the current value for the **Output queue (OUTQ)** parameter is *DEV or *WRKSTN.

Top

CL var for TEXT (50) (TEXT)

Specifies the name of a variable that is used to retrieve the user-defined description for the specified user profile. In CL programs, this should be a 50-character variable. If there is no text associated with the user, blanks are returned in the CL variable.

CL var for PWDCHGDAT (6) (PWDCHGDAT)

Specifies the name of a variable that is used to retrieve the date when the password for the specified user was last changed. The date is returned in the form YYMMDD. In CL programs, this should be a 6-character variable. If the user does not have a date, blanks are returned.

CL var for USRCLS (10) (USRCLS)

Specifies the name of a variable that is used to retrieve the user class for the specified user. A special value of *USER, *SYSOPR, *PGMR, *SECADM, or *SECOFR is returned in the variable. In CL programs, this should be a 10-character variable.

CL var for ASTLVL (10) (ASTLVL)

Specifies the name of a variable that is used to retrieve the assistance level for the specified user. A special value of *SYSVAL, *BASIC, *INTERMED, or *ADVANCED is returned in the variable. In CL programs, this should be a 10-character variable.

CL var for SPCENV (10) (SPCENV)

Specifies the name of a variable that is the starting environment for the specified user. A special value of *SYSVAL, *NONE, or *S36 is returned in the variable. In CL programs, this should be a 10-character variable.

CL var for CURLIB (10) (CURLIB)

The name of the variable that is used to retrieve the name of the job's default library for the specified user. A value of *CRTDFT is returned in the variable if no current library exists for this user. In CL programs, this should be a 10-character variable.

CL var for INLMNU (10) (INLMNU)

Specifies the name of a variable that is used to retrieve the name of the initial menu that is shown when the specified user signs on to the system. In CL programs, this should be a 10-character variable.

CL var for INLMNULIB (10) (INLMNULIB)

The name of the variable that is used to retrieve the library name that contains the initial menu. In CL programs, this should be a 10-character variable.

CL var for LMTCPB (10) (LMTCPB)

Specifies the name of a variable that is used to retrieve the values for the limits to which users can change their user profiles and run commands. In CL programs, this should be a 10-character variable. A special value of *NO, *YES, or *PARTIAL is returned in the variable.

CL var for DLVRY (10) (DLVRY)

Specifies the name of a variable that is used to retrieve the message control delivery value for the specified user profile. In CL programs, this should be a 10-character variable. The special value of *NOTIFY, *BREAK, *HOLD, or *DFT is returned in the variable.

CL var for SEV (2 0) (SEV)

Specifies the name of a variable that is used to retrieve the message control severity level for the specified user. In CL programs this should be a variable length of (2 0).

CL var for PRTDEV (10) (PRTDEV)

Specifies the name of a variable that is used to retrieve the name of the printer device for the specified user. In CL programs, this should be a 10-character variable. A value of *SYSVAL is returned if the printer device name is from the system value QPRTDEV. A value of *WRKSTN is returned if the printer device name is from the printer device assigned to the user's work station.

CL var for ATNPGM (10) (ATNPGM)

Specifies the name of a variable that is used to retrieve the name of the Attention key handling program for the specified user. In CL programs, the variable has a length of 10 characters. A value of *SYSVAL if the attention key handling program is from system value QATNPGM. A value of *NONE is returned if no Attention key handling program was specified for this user.

CL var for ATNPGMLIB (10) (ATNPGMLIB)

Specifies the name of a variable that is used to retrieve the name of the Attention key handling program library for the specified user. In CL programs, this should be a 10-character variable. If *NONE is the current value for the **Attention program (ATNPGM)** parameter, blanks are returned in the variable.

CL var for USROPT (240) (USROPT)

Specifies the name of a variable that is used to retrieve the list of values for user options for the specified user. In CL programs, this should be a 240-character variable. The special value of *NONE or a list of values is returned in the variable.

CL var for DSPSGNINF (7) (DSPSGNINF)

Specifies the name of a variable that is used to retrieve the sign-on information display indicator for the specified user. In CL programs, this should be a 7-character variable. The special value of *SYSVAL, *YES, or *NO is returned in the variable. If *SYSVAL is returned, the display sign-on information indicator is from the system value QDSPSGNINF.

CL var for PWDEXPITV (5 0) (PWDEXPITV)

Specifies the name of a variable that is used to retrieve the password expiration interval for the specified user. In CL programs, the variable specified must be packed (5,0) in length. The value returned is either a number ranging from 1 through 366, 0 if it is *SYSVAL, or -1 if it is *NOMAX. If 0 is returned (*SYSVAL), then the password expiration interval is from the system value QPWDEXPITV.

CL var for PWDEXP (4) (PWDEXP)

Specifies the name of a variable that is used to retrieve the password expired indicator for the specified user. In CL programs, this should be a 4-character variable. The special value of *YES or *NO is returned in the variable.

CL var for STATUS (10) (STATUS)

The name of a variable that is used to retrieve the status of the specified user profile. In CL programs, this should be a 10-character variable. The special value of *ENABLED or *DISABLED is returned in the variable.

CL var for PRVSIGN (13) (PRVSIGN)

Specifies the name of a variable that is used to retrieve the previous sign-on date and time for the specified user. The date and time are returned in the form CYYMMDDHHMMSS. In CL programs, this should be a 13-character variable. If the user has not signed on previously, blanks are returned.

CL var for NOTVLDSIGN (11 0) (NOTVLDSIGN)

Specifies the name of a variable that is 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.

CL var for LMTDEVSSN (7) (LMTDEVSSN)

Specifies the name of a variable that is used to retrieve the limit device sessions indicator for the specified user. The special value of *SYSVAL, *YES, or *NO is returned in the CL variable. If *SYSVAL is returned, the limit device sessions indicator is from the system value QLMTDEVSSN. In CL programs, this should be a 7-character variable.

CL var for KBDBUF (10) (KBDBUF)

Specifies the name of a variable that is used to retrieve the keyboard buffering value for the specified user. The special value of *SYSVAL, *NO, *TYPEAHEAD, or *YES is returned in the CL variable. If *SYSVAL is returned, the keyboard buffering value is the same as the system value QKBDBUF. In CL programs, this should be a 10-character variable.

CL var for LANGID (10) (LANGID)

Specifies the name of a variable that is used to retrieve the language identifier for the specified user. The special value *SYSVAL or the language identifier is returned in the variable. If *SYSVAL is returned, the language identifier for the user is determined by the QLANGID system value. In CL programs, this should be a 10-character variable.

CL var for CNTRYID (10) (CNTRYID)

Specifies the name of a variable that is used to retrieve the country or region identifier for the specified user. The special value *SYSVAL or the country or region identifier is returned in the variable. If *SYSVAL is returned, the country or region identifier for the user is determined by the QCNTRYID system value. In CL programs, this should be a 10-character variable.

CL var for CCSID (5 0) (CCSID)

Specifies the name of a variable that is used to retrieve the coded character set identifier (CCSID) for the specified user. A 5-digit value is returned or, if the system value QCCSID is used to determine the CCSID for the user, a value of -2 is returned. In CL programs, this should be a decimal variable length of (5 0).

CL var for SRTSEQ (10) (SRTSEQ)

Specifies the name of a variable used to retrieve the sort sequence table for the specified user. The value returned is one of the following: *HEX, *LANGIDUNQ, *LANGIDSHR, *SYSVAL, or the 10-character table identifier. If *SYSVAL is returned, the table identifier for the user is determined by the QSRTSEQ system value.

CL var for SRTSEQLIB (10) (SRTSEQLIB)

Specifies the name of a variable used to retrieve the sort sequence table library for the specified user. The value returned is the 10-character library identifier. The variable is set to blanks unless a sort sequence table name is specified.

CL var for OBJAUD (10) (OBJAUD)

The name of a 10-character variable that is used to retrieve the object auditing value for the specified user. The special value of *NONE, *CHANGE, or *ALL, as specified on the Change User Audit (CHGUSRAUD) command, is returned in the variable. When the user of this command does not have either *ALLOBJ or *AUDIT special authority, *NOTAVL is returned. When less than 10 characters are returned, the variable is padded on the right with blanks.

CL var for AUDLVL (640) (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, *JOBDTA, *OBJMGT, *OFCSRV, *PGMADP, *SAVRST, *SECURITY, *SERVICE, *SPLFDTA, or *SYSMGT. If the user of this command does not have either *ALLOBJ or *AUDIT special authority, the first entry contains the value *NOTAVL followed by blanks.

CL var for GRPAUTTYP (10) (GRPAUTTYP)

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.

CL var for SUPGRPPRF (150) (SUPGRPPRF)

Specifies the name of a variable used to retrieve the supplemental group profiles for the specified user. In CL programs, the variable has a length of 150 characters. The format returned is a list of up to 15 supplemental group profiles, with each entry 10 characters long. If there are fewer than 15 supplemental groups in the list, the remaining entries are padded on the right with blanks. If the user has no supplemental groups, the first entry contains the value of *NONE followed by blanks.

The supplemental group profiles are returned in the following format:

Entry-1
   Supplemental group profile CHAR(10)
Entry-2
  Supplemental group profile CHAR(10)
   .
   .
   .
Entry-15
Supplemental group profile CHAR(10)

<div align="right">Top</div>

CL var for UID (10 0) (UID)

Specifies the name of a variable used to retrieve the user ID number (uid) for the specified user. In CL programs, this should be a decimal variable length of (10 0).

<div align="right">Top</div>

CL var for GID (10 0) (GID)

Specifies the name of a variable used to retrieve the group ID number (gid) for the specified user. In CL programs, this should be a decimal variable length of (10 0). The value returned is either a 10-digit value or a value of 0 if the gid is *NONE. The variable must be a 10-digit value with no decimal positions.

<div align="right">Top</div>

CL var for SETJOBATR (160) (SETJOBATR)

Specifies the name of a variable used to retrieve the job attributes set from the LOCALE path name of the specified user. In CL programs, the variable has a length of 160 characters. The format returned is a list of up to 16 job attributes that are set, with each entry 10 characters long. If there are fewer than 16 attributes in the list, the remaining entries are padded on the right with blanks. If no attributes are set from the locale path name for the user, the first entry contains the value of *NONE followed by blanks.

<div align="right">Top</div>

Character identifier control (CHRIDCTL)

Specifies the name of a variable used to retrieve the character identifier control for the specified user. In CL programs, this variable should have a length of 10 characters. The special value of *DEVD, *JOBCCSID, or *SYSVAL is returned in the variable. If *SYSVAL is returned, the CHRID control for the user is determined by the QCHRIDCTL system value.

<div align="right">Top</div>

CL var for LCLPWDMGT (10) (LCLPWDMGT)

Specifies the name of a variable used to retrieve local password management for the specified user. In CL programs, this variable should have a length of 10 characters. The special value of *YES or *NO is returned in the variable.

Examples

```
CRTUSRPRF   USRPRF(SMITH)  SPCAUT(*SAVSYS *SECADM)
            MAXSTG(*NOMAX)  PTYLMT(4)
            INLPGM(*NONE)  MSGQ(QGPL/SMITHMQ)
            OUTQ(QGPL/QSMITH)  TEXT('John Smith User Profile')
```

If the SMITH user profile is created using the above command, when user SMITH calls a CL program containing the following:

```
DCL   &UNAME    *CHAR  10
DCL   &URIGHT   *CHAR  100
DCL   &IPGM     *CHAR  10
DCL   &IPGMLB   *CHAR  10
DCL   &UMSGQ    *CHAR  10
DCL   &UMSQLB   *CHAR  10
DCL   &USED     *CHAR  10
 :
RTVUSRPRF   USRPRF(*CURRENT)  STGUSED(&USED)  +
            RTNUSRPRF(&UNAME)  SPCAUT(&URIGHT)  +
            INLPGM(&IPGM)  INLPGMLIB(&IPGMLB)
```

This command retrieves the user profile information for the job's current user profile and returns the information into the following CL program variables:

```
&UNAME     'SMITH        '
&URIGHT    |'*SAVSYS    *SECADM    (        ) ... |
           |_____100 characters_____|
&IPGM      '*NONE        '
&IPGMLB    '             '
```

Error messages

*ESCAPE Messages

CPF2203

> User profile &1 not correct.

CPF2204

> User profile &1 not found.

CPF2213

> Not able to allocate user profile &1.

CPF2217

> Not authorized to user profile &1.

CPF2225

> Not able to allocate internal system object.

CPF8134

> User profile &4 damaged.

Retrieve User Print Info (RTVUSRPRTI)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

The Retrieve User Print Information (RTVUSRPRTI) command is used in a CL program to retrieve the user print information value associated with a user profile. The values are returned in the specified CL variables for the desired user.

Parameters

Keyword	Description	Choices	Notes
USER	User	*Name*, *CURRENT	Required, Positional 1
RTNTEXT	CL var for RTNTEXT	*Character value*	Required, Positional 2

User (USER)

Specifies the user whose user print information is being retrieved.

This is a required parameter.

*CURRENT
>   The user profile under which the current job is running is used.

name   Specify the name of the user whose user print information is being retrieved.

CL var for RTNTEXT (RTNTEXT)

Specifies a 10-character CL variable used to retrieve the user print information of the user profile for which information is requested.

This is a required parameter.

Examples

```
RTVUSRPRTI   USER(FEIST)   RTNTEXT(&TEXT)
```

This command retrieves user print information for user profile FEIST and stores it in CL program variable &TEXT.

Error messages

*ESCAPE Messages

CPF2204

User profile &1 not found.

CPF2217

Not authorized to user profile &1.

CPF2247

Internal security object not available. Reason code &1.

CPF34D5

CCSID translation error.

Retrieve WSCST source (RTVWSCST)

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

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.

Parameters

Keyword	Description	Choices	Notes
DEVTYPE	Device type	*TRANSFORM, 3101, 3151, 3161, 3162, 3163, 3164, 3179, 3180, 3196, 3197, 3476, 3477, 3486, 3487, 4019, 4201, 4202, 4207, 4208, 4216, 4224, 4234, 5204, 5251, 5291, 5292, 6252, D220, T910, T925, T955, V100, V220, W30, W50, W60	Required, Positional 1
KBDTYPE	Keyboard language type	AGB, AGE, AGI, ALI, BGB, BLI, BLM, BRB, BRE, CAB, CAI, CAM, CLB, CSB, CYB, DMB, DME, DMI, FAB, FAE, FAI, FNB, FNE, FNI, FQB, FQI, GKB, GNB, HNB, ICB, ICE, ICI, INB, INI, ITB, ITE, IRB, ITI, JEB, JEI, JPB, KAB, MKB, NCB, NEB, NEE, NEI, NWB, NWE, NWI, PLB, PRB, PRE, PRI, RMB, ROB, RUB, SFI, SFM, SGI, SGM, SKB, SPB, SPE, SQB, SPI, SSB, SSE, SSI, SWB, SWE, SWI, THB, TKB, TRB, UKB, UKE, UKI, USB, USE, USI, YGI	Optional, Positional 2
MFRTYPMDL	Manufacturer type and model	*Character value*	Optional
SRCMBR	Source member	*Name*	Optional, Positional 3
KBD	Keyboard attached	*DATA5250, *TYPE5250, *DATA122, *TYPE122, *ENHANCED	Optional
SRCFILE	Source file	*Qualified object name*	Optional
	Qualifier 1: Source file	*Name*, **QTXTSRC**	
	Qualifier 2: Library	*Name*, *CURLIB, ***LIBL**	
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Device type (DEVTYPE)

Specifies the device type.

The possible values are:

*TRANSFORM
> The SCS-to-ASCII host print transform function support is used by the ASCII printer.

device-type
> Specify the device type to be used. See the Workstation Customization Programming book for a list of allowed device types.

Keyboard language type (KBDTYPE)

Specifies the 3-character keyboard language identifier (used for EBCDIC and ASCII) for this display station.

The possible value is:

keyboard-language-type
>   Specify the 3-character country or region identifier (used for EBCDIC and ASCII) for this display station.

>   See the **Keyboard language type** prompt (KBDTYPE parameter) on the Change Device Description (Display) (CHGDEVDSP) or the Create Device Description (Display) (CRTDEVDSP) in the CL Reference for a list of the valid identifiers and the language the identifier represents. The ASCII device groups (if applicable) are also shown for each language.

Manufacturer type and model (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 in the CL Reference for a list of the supported manufacturers, types, and models for ASCII printers using host print transform function support.

Source member (SRCMBR)

Specifies the name of the source file member to receive the retrieved table attributes.

Keyboard attached (KBD)

Specifies the keyboard type.

The possible values are:

*DATA5250
>   A 5250 data entry keyboard is specified.

*TYPE5250
>   A 5250 typewriter keyboard is specified.

*DATA122
>   A 122 key data entry keyboard is specified.

*TYPE122
>   A 122 key typewriter keyboard is specified.

*ENHANCED
>   An enhanced keyboard is specified.

Source file (SRCFILE)

Specifies the name of the source file in which a member is created to contain the retrieved table attributes. If the source file does not exist, it is created. The coded character set identifier for the source file is *HEX.

The possible library values are:

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

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

library-name
>   Specify the name of the library where the source file is located.

The possible values are:

QTXTSRC
>   The IBM-supplied source file QTXTSRC is used.

source-file-name
>   Specify the name of the source file.

<div align="right">Top</div>

Text 'description' (TEXT)

Specifies the description of the created source physical file member.

The possible values are:

*BLANK
>   Text is not specified.

'description'
>   Specify a description for the source physical file member.

<div align="right">Top</div>

Examples

```
RTVWSCST   DEVTYPE(5251)  KBDTYPE(USB)  SRCMBR(MYSOURCE)
           KBD(*DATA5250)  SRCFILE(MYLIB/QTXTSRC)
```

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 QTXTSRC in library MYLIB.

<div align="right">Top</div>

Error messages

*ESCAPE Messages

CPF5D33

Request for system table failed. Reason code &4.

Run Backup (RUNBCKUP)

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

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.

Parameters

Keyword	Description	Choices	Notes
BCKUPOPT	Backup options	*DAILY, *WEEKLY, *MONTHLY	Required, Key, Positional 1
DEV	Device	Single values: ***BCKUPOPT** Other values (up to 4 repetitions): *Name*	Optional, Positional 2

Backup options (BCKUPOPT)

Specifies the backup options to use.

This is a required parameter.

*DAILY
> The daily backup options are used.

*WEEKLY
> The weekly backup options are used.

*MONTHLY
> The monthly backup options are used.

Device (DEV)

Specifies a list of tape devices to use for the backup.

Single values

*BCKUPOPT
> The tape device names stored in the specified options are used for the backup.

Other values (up to 4 repetitions)

name   Specify a list of tape devices used for the backup. If you are using a virtual tape device you can only specify one device name.

Examples

Example 1: Running a Daily Backup

```
RUNBCKUP   BCKUPOPT(*DAILY)
```

This command runs the daily backup using the devices specified in the options.

Example 2: Running a Monthly Backup

```
RUNBCKUP   BCKUPOPT(*MONTHLY)  DEV(TAP02)
```

This command runs the monthly backup using device TAP02 instead of those specified in the options.

Error messages

*ESCAPE Messages

CPF1EE3
> Not authorized to backup options.

CPF1EE4
> Not authorized to run backup.

CPF1EE6
> Devices specified cannot be used.

CPF1EE7
> Unexpected error occurred during backup.

CPF1EE8
> Unexpected error from exit program &2/&1.

CPF1E6C
> Backup options in use.

CPF1E6E
> Nothing selected for backup.

CPF1E63
> Error occurred while checking backup authority.

CPF1E67
> Backup options and library backup list damaged.

CPF1E68
> Backup incomplete.

CPF1E99
> Unexpected error occurred.

Run LPDA-2 (RUNLPDA)

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

The Run LPDA-2 (RUNLPDA) command allows you to run a Link Problem Determination Aid-2 (LPDA-2) operational command on local or remote data circuit-terminating equipment (DCE). The RUNLPDA command can be used to:

- Establish or disconnect a switched telephone network connection.
- Open or close the relay contact in a coupler.
- Determine whether a relay contact is open or closed.
- Determine whether electric current is flowing through an internal sensor.
- Change the transmit speed of a DCE to full or backup.

The result of the RUNLPDA command is returned as a message.

Restrictions:

1. The RUNLPDA command is valid only for an analog LPDA-2 DCE attached to a nonswitched SDLC line.
2. This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, or QSRVBAS, user profiles have private authorities to use the command.

Parameters

Keyword	Description	Choices	Notes
LINE	Line	*Name*	Required, Positional 1
OPTION	Option	*CALL, *DSC, *SETSPEED, *CONTSENSE, *CONTOPER	Required, Positional 2
LCLDCEADR	Local DCE address	01-FB, ***LCL**	Optional
RMTDCEADR	Remote DCE address	01-FB, ***NONE**, *ANY	Optional
TELNBR	Telephone number	Single values: ***STORED** Other values: *Element list*	Optional
	Element 1: Number 1	*Character value*	
	Element 2: Number 2	*Character value*	
CONTACT	Relay contact	*OPEN, *CLOSE	Optional
SPEED	Transmit speed	*FULL, *BACKUP	Optional
DTEPORT	DTE port	***ALL**, A, B, C, D	Optional
DCERTY	DCE retry	***NO**, *YES	Optional

Line (LINE)

Specifies the name of the nonswitched SDLC line that is attached to the analog DCE on which the LPDA-2 operational command is to be run. If *CALL is specified on the **Option** prompt (OPTION parameter), the line must be varied on but not active. If *CONTOPER, *CONTSENSE, *DSC, or *SETSPEED is specified on the OPTION parameter, the line must be either varied on or active.

This is a required parameter.

Option (OPTION)

Specifies which LPDA-2 operational command is run.

This is a required parameter.

The contact operate, contact sense, and set transmit speed commands can be run on a local DCE or a remote DCE.

To run one of these commands on a local DCE:
- On the **Local DCE address** prompt (LCLDCEADR parameter), specify the address of the local DCE.
- On the **Remote DCE address** prompt (RMTDCEADR parameter), specify *NONE.

To run one of these commands on a remote DCE:
- On the LCLDCEADR parameter, specify the address of the local DCE to which the remote DCE is connected.
- On the RMTDCEADR parameter, specify the address of the remote DCE.

Note: If the local DCE is configured as point-to-point secondary or multipoint tributary, LPDA-2 commands are not sent to the remote DCE.

The possible values are:

*CALL
> The call out command is run. This command establishes a connection between a local and a remote DCE over a switched telephone network. This value is valid only if:
> - Both the local DCE and the remote DCE have two-wire couplers installed or both the local DCE and the remote DCE have four-wire couplers installed.
> - The line specified on the **Line** prompt (LINE parameter) is varied on, but not active.

*CONTOPER
> The contact operate command is run. This command opens or closes the relay contact in the coupler, depending on the value specified on the **Relay contact** prompt (CONTACT parameter). This value is valid only if a two-wire coupler is installed in the DCE on which this LPDA-2 command is run.

*CONTSENSE
> The contact sense command is run. This command reports whether the relay contact in the coupler is open or closed and whether electric current is flowing through the internal sensor. This option is valid only if a two-wire coupler is installed in the DCE on which this command is run.

*DSC The disconnect command is run. This command disconnects the switched telephone network connection between the local DCE and the remote DCE. If you specify this value, you cannot specify *NONE on the **Remote DCE address** prompt (RMTDCEADR parameter).

*SETSPEED

> The set transmit speed command is run. This command changes the transmit speed of the DCE to full or backup, depending on the value specified on the **Transmit speed** prompt (SPEED parameter). For multiple port DCE configurations in which the data terminal equipment (DTE) ports can be set to different speeds, use the **DTE port** prompt (DTEPORT parameter) to specify the port.
>
> **Note:** The set transmit speed command may not change the transmit speed, depending on the configuration options selected for the DCE. Refer to the DCE documentation for more information.

<div align="right">Top</div>

Local DCE address (LCLDCEADR)

Specifies the hexadecimal address of the local DCE. Refer to the DCE documentation for more information on addressing.

The possible values are:

*LCL   X'01' is used for the address.

local-DCE-address
> Specify the address of the local DCE. Valid values range from X'01' through X'FB'.

<div align="right">Top</div>

Remote DCE address (RMTDCEADR)

Specifies the hexadecimal address of the remote DCE on which the LPDA-2 operational command is to be run.

The possible values are:

*NONE
> X'00' is used as the address, which indicates that the LPDA-2 command is to be run on the local DCE. You cannot specify *NONE on this parameter if you specify *DSC on the **Option** prompt (OPTION parameter).

*ANY   X'FD' is used for the address. Specify this value in the following situations:

- The LPDA-2 command is to be run on any remote DCE connected to the local DCE.
- You do not know the remote DCE address on a point-to-point line.
- To run the LPDA-2 operational command on all tributary DCEs on a multipoint line. In this case, no detailed response is received.

remote-DCE-address
> Specify the address of the remote DCE. Valid values range from X'01' through X'FB'.

<div align="right">Top</div>

Telephone number (TELNBR)

Specifies the telephone number or telephone numbers that the local DCE dials to establish a connection to the remote DCE. This parameter is valid only if *CALL is specified on the **Option** prompt (OPTION parameter).

The possible values are:

*STORED

        The telephone number or numbers stored in the local DCE when the DCE was configured are used.

telephone-number

        Specify one telephone number if a two-wire coupler is installed in the local DCE. Specify two telephone numbers if a four-wire coupler is installed in the local DCE. Only numeric characters are processed by the LPDA-2 command, but you can also enter alphabetic characters or any other non-DBCS characters to improve readability. A comma (,) can be used to instruct the DCE to pause during dialing.

Relay contact (CONTACT)

Specifies whether to open or close the relay contact in a two-wire coupler. This parameter is required if *CONTOPER is specified on the **Option** prompt (OPTION parameter).

The possible values are:

*OPEN

        The relay contact is opened.

*CLOSE

        The relay contact is closed.

Transmit speed (SPEED)

Specifies desired transmit speed of the DCE. This parameter is required if *SETSPEED is specified on the **Option** prompt (OPTION parameter).

The possible values are:

*FULL The transmit speed is set to full.

*BACKUP

        The transmit speed is set to backup.

DTE port (DTEPORT)

Specifies the DTE port on the local or remote DCE for which the transmit speed is changed. This parameter is valid only if *SETSPEED is specified on the **Option** prompt (OPTION parameter) and is applicable only to multiple port DCEs that do not use the multiple address configuration option.

The possible values are:

*ALL The aggregate speed of the DCE is changed. Refer to the DCE documentation for information on the effect of changing the aggregate speed on the transmit speed of individual ports.

A   The transmit speed of the A-port is changed.

B   The transmit speed of the B-port is changed.

C   The transmit speed of the C-port is changed.

D   The transmit speed of the D-port is changed.

DCE retry (DCERTY)

Specifies whether the local DCE resends the LPDA-2 command to the remote DCE if no response is received from the remote DCE. No retry can be attempted if *CALL or *DSC is specified on the **Option** prompt (OPTION parameter).

The possible values are:

*NO   No retry is attempted.

*YES   One retry is attempted.

Examples

Example 1: Establishing a Switched Telephone Network Connection

```
RUNLPDA   LINE(SDLCLINE)  OPTION(*CALL)  LCLDCEADR(*LCL)
          RMTDCEADR(*NONE)  TELNBR(*STORED)
```

This command runs the call out command. The local DCE with address X'01' (*LCL) on line SDLCLINE dials the telephone numbers that are stored in the local DCE.

Example 2: Establishing a Switched Telephone Network Connection

```
RUNLPDA   LINE(SDLCLINE)  OPTION(*CALL)
          LCLDCEADR(*LCL)  RMTDCEADR(*NONE)
          TELNBR('9, 1-507-555-1212' '9, 1 (507) 555-1313')
```

This command runs the call out command. The local DCE dials the two numbers specified on the TELNBR parameter. The comma (,) indicates a pause during dialing. Other non-numeric characters are ignored, but are allowed for easier reading.

Example 3: Disconnecting a Switched Telephone Network Connection

```
RUNLPDA   LINE(SDLCLINE)  OPTION(*DSC)
          LCLDCEADR(10)  RMTDCEADR(*ANY)
```

This command runs the disconnect command. The local DCE with address X'10' disconnects from the switched telephone network.

Example 4: Closing the Relay Contact in the Local DCE

```
RUNLPDA   LINE(SDLCLINE)  OPTION(*CONTOPER)  LCLDCEADR(02)
          RMTDCEADR(*NONE)  CONTACT(*CLOSE)
```

This command runs the contact operate command. The local DCE with address X'02' closes the relay contact in its two-wire coupler.

Example 5: Reporting the Status of the Relay Contact

```
RUNLPDA   LINE(SDLCLINE)  OPTION(*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

*ESCAPE Messages

CPF1BAA
> LPDA-2 command cannot run in switched network backup.

CPF1BAB
> LPDA-2 command not supported by target DCE.

CPF1BAC
> Required feature not installed.

CPF1BAD
> LPDA-2 command not compatible with DCE configuration.

CPF1BA1
> Line description &1 does not exist.

CPF1BA2
> Line &1 not active or not varied on.

CPF1BA3
> Line &1 is active secondary.

CPF1BA4
> Line &1 failed while processing RUNLPDA.

CPF1BA5
> No response received for LPDA-2 command.

CPF1BA6
> Cannot run LPDA-2 command on line &1.

CPF1BA7
> No buffers available for RUNLPDA data.

CPF1BA8
> LPDA-2 command already active on line &1.

CPF1BBA
> Continuous answer tone received for first call.

CPF1BBB
> Continuous answer tone received for second call.

CPF1BBC
> No line signal on switched connection.

CPF1BBD

No initial dial tone.

CPF1BBE

No dial tone after pause.

CPF1BBF

Busy tone received from remote DCE.

CPF1BB0

Coupler not installed in local DCE.

CPF1BB1

Coupler not operational.

CPF1BB2

LPDA-2 command not valid with configuration options.

CPF1BB3

Telephone number not valid.

CPF1BB5

Switched connection already active.

CPF1BB6

Only single telephone number allowed.

CPF1BB7

Two telephone numbers required.

CPF1BB8

No answer from remote DCE on first call.

CPF1BB9

No answer from remote DCE on second call.

CPF1BDA

Error occurred processing RUNLPDA command.

CPF1BDB

Internal processing error.

CPF1BDC

Required feature not operational.

CPF1BDD

Ring back limit exceeded.

CPF1BDE

DCE busy. LPDA-2 command cannot be run.

CPF1BDF

Line not in proper state for LPDA-2 call out command.

CPF1BD3

DCE currently processing previous LPDA-2 command.

CPF1BD5

Cannot send command to remote DCE.

CPF1BD6

Line description &1 not nonswitched *SDLC.

CPF1BD8

Request failed. System service tools active.

CPF1BD9

       RUNLPDA command does not support switched lines.

CPF1B85

       Two-wire coupler not installed.

CPF1B86

       Coupler not installed.

CPF1B87

       Transmit speed cannot be changed.

CPF1B88

       DTE port selected not valid for configuration.

CPF1B94

       Local DCE received no response from remote DCE.

CPF1B95

       Not authorized to line description &1.

CPF1B97

       Format of response received not valid.

CPF1B98

       Received response with bad frame check sequence.

OFC8DB7

       Internal processing failure.

Top

Run Query (RUNQRY)

The Run Query (RUNQRY) command runs an existing query or a default query if only a file name is specified for this command. The query gets information from the system database and produces a report of that information. The report is created in either detailed or summary form. The definition of the query can be printed when output to a printer or database file is specified. The output is shown, printed, or stored in a database file. The command is used in three ways: to run an existing query (one that has already been created), to run an existing query with some of its values changed by values specified on this command, or to run a default query based only on the defaults and values specified in this command.

- To run an existing query without changing the file or files to query, use the QRY parameter (without the QRYFILE parameter) to specify the name of the query.
- To run a changed version of an existing query, use the QRY parameter and the appropriate parameters to change the definition as desired. The parameter values specified on this command override the corresponding values in the existing query definition, but only when the command is processing. For example, use the QRYFILE parameter to indicate a different file or list of files to use in the query.
- To query a file without a previously defined query definition, use the QRYFILE parameter to specify which file to query. Only one file name can be specified for a default query.

If you specify both the QRY and QRYFILE parameters, the files specified in the QRYFILE parameter override the file names specified in the query. Therefore, if multiple files (and members) are defined in the query definition and you want to change one or two of them, specify *SAME for the file selections that do not change, and specify the values for the files to override.

Notes:

1. When a changed version of an existing query is run, the changes specified on the RUNQRY command do **not** change any of the values in the query definition itself; they affect only the results of the report being run.
2. If an existing query (identified on the QRY parameter) is used, *RUNOPT is the default value for any unspecified parameters; that is , the same value specified (or assumed) in the definition of the query is used as the default. If this is a default query, the default value is not *RUNOPT, but is the next predefined value listed in the syntax diagram.

Top

Parameters

Keyword	Description	Choices	Notes
QRY	Query	Single values: *NONE Other values: *Qualified object name*	Optional, Positional 1
	Qualifier 1: Query	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Keyword	Description	Choices	Notes
QRYFILE	Query file	Values (up to 32 repetitions): *Element list*	Optional, Positional 2
	Element 1: File	*Qualified object name*	
	Qualifier 1: File	*Name*, *SAME	
	Qualifier 2: Library	*Name*, *RUNOPT, ***LIBL**, *CURLIB	
	Element 2: Member	*Name*, *RUNOPT, ***FIRST**, *LAST, *ALL	
OUTTYPE	Report output type	***RUNOPT**, *DISPLAY, *PRINTER, *OUTFILE	Optional, Positional 3
OUTFORM	Output form	***RUNOPT**, *DETAIL, *SUMMARY	Optional
RCDSLT	Record selection	***NO**, *YES	Optional
PRTDFN	Print query definition	***RUNOPT**, *NO, *YES	Optional
PRTDEV	Print device	*Name*, ***RUNOPT**, *PRINT	Optional
FORMSIZE	Form size	*Element list*	Optional
	Element 1: Form length	1-255, ***RUNOPT**	
	Element 2: Form width	1-378, ***RUNOPT**	
FORMTYPE	Form type	*Character value*, ***RUNOPT**, *STD	Optional
COPIES	Copies	1-255, ***RUNOPT**	Optional
LINESPACE	Line spacing	***RUNOPT**, 1, 2, 3	Optional
OUTFILE	Output file	*Element list*	Optional
	Element 1: File	*Qualified object name*	
	Qualifier 1: File	*Name*, ***RUNOPT**	
	Qualifier 2: Library	*Name*, ***RUNOPT**, *CURLIB	
	Element 2: Member	*Name*, *RUNOPT, ***FIRST**, *LAST, *ALL	
	Element 3: Option	***RUNOPT**, *NEWFILE, *RPLFILE, *NEWMBR, *RPLMBR, *ADDMBR	
AUT	Authority	*Name*, ***RUNOPT**, *LIBCRTAUT, *USE, *CHANGE, *ALL, *EXCLUDE	Optional

Top

Query (QRY Parameter)

Specifies the name of an existing query to be run. If QRY is not specified, QRYFILE must be specified.

The possible values are:

*NONE
> No existing query definition is used. Instead, a default query (or quick query) is used to get information from the file specified on the QRYFILE parameter.

query-name
> Specify the name of the query to run.

> If no value is specified, the file or files that were specified when the query was defined are the files to be queried.

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

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

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

library-name
> Specify the library where the query is located.

Query file (QRYFILE Parameter)

Specifies the database file or files to be queried for information. If the QRY parameter is specified, as many as 32 files can be specified on this parameter by using the file names and/or using the default value *SAME for one or more of the files. If the QRY parameter is not specified, only one file name can be specified on this parameter. If QRYFILE is not specified, QRY must be specified.

If no value is specified, the file or files that were specified when the query was defined are the files used to run the query.

You can enter multiple values for this parameter.

There are two parts to this parameter.

Element 1: Database File Name

The possible values are:

data-base-file-name
> Specify the names of one or more database files that contain the data from which the system gets information to produce the output. Up to 32 files can be specified.

*SAME
> The list of files remains the same as defined in the query definition.

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

*SAME
> The value of the library, file, and member do not change for the file in this position within the list. The values are the same as defined in the query. Any values specified for the library, file, and member are ignored.

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

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

*RUNOPT
> The library specified for the file selection is used.

library-name
> Specify the library where the database file is located.

Element 2: Database File Member

The possible values for member name are:

member-name
> Specify the name of the file member to be queried.

*RUNOPT
> The member specified for this file selection is used.

*FIRST

>   The first member in the file is the member to be queried.

*LAST

>   The last member in the file is the member to be queried.

*ALL   The input file is a partitioned table, where all of the members in the file are to be queried.

Report output type (OUTTYPE Parameter)

Specifies where the report or output produced by the query is sent. If a value is not specified in the query, and is not entered on the command, or if a query name is not specified, *DISPLAY is assumed.

The possible values are:

*RUNOPT

>   If a query definition is being used, the type of output specified in the query definition is the type of output produced when this query is run.

*DISPLAY

>   The output produced by the query is sent to the display station that runs the command. If run in batch, the output is sent to the printer instead of the display.

*PRINTER

>   The output produced by the query is printed.

*OUTFILE

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

Output form (OUTFORM Parameter)

Specifies the form of output produced by the query. If no value was specified in the query and no value was entered on the command, or if a query name is not specified, *DETAIL is assumed.

The possible values are:

*RUNOPT

>   If a query definition is used, the output form specified in the query definition is used when this query is run.

*DETAIL

>   The output form produced by the query is a report containing detail records and summary records if any exist.

*SUMMARY

>   The output form produced by the query is a report containing summary records only.

Record selection (RCDSLT Parameter)

Specifies whether or not the query is run with a run time selection test.

Note: The Query for iSeries licensed program must be installed and the query must be run interactively to specify *YES.

The possible values are:

*NO   The query is run without showing the record selection display in Query.

*YES   Record selection definition is allowed for this run only. A display is shown on which you can change the record selection tests defined in the query or specify record selection tests if a query name was not specified.

        Note: For output to display, query can be run repeatedly.

Print query definition (PRTDFN Parameter)

Specifies whether or not the query definition is printed with the report when the query is run. The definition can be printed when the output of the query is printed or is stored in a database file, as determined by the OUTTYPE parameter. If a value is not specified in the query, or in this parameter, or if a query name is not specified, the value *NO is assumed.

The possible values are:

*RUNOPT
>If a query definition is being used when the query is run, the print option specified in the query definition is used.

*NO   The query definition is not printed when the query is run.

*YES   The query definition is printed in the report. *YES cannot be specified if OUTTYPE(*DISPLAY) is specified or assumed.

Print device (PRTDEV Parameter)

Specifies the printer device on which the report is printed. If no value is specified, the printer that was specified when the query was defined is assumed. If no printer is specified in the query or in this parameter, or if a query name is not specified, the value *PRINT is assumed.

Note: If an override is in effect for the printer file QPQUPRFIL, this parameter uses the value specified by the override.

The possible values are:

*RUNOPT
>If a query definition is being used, the printer specified in the query definition is used to print the output when this query is run.

*PRINT
>The default printer, as defined by QPQUPRFIL, is used to print the output when this query is run.

printer-device-name
>Specify the name of the printer that is used to print the output when this query is run.

Form size (FORMSIZE Parameter)

Specifies the length and the width of the forms on which the report is printed. If a form width greater than 132 is specified, Query will open the file with CPI(15) and FONT(*CPI). If no value is specified in the query or in this parameter, or if a query name is not specified, 132 is the assumed form width, and the value from the file QPQUPRFIL is the assumed form length.

Note: If an override is in effect for the printer file QPQUPRFIL, this parameter uses the value specified by the override.

There are two parts to this parameter.

Element 1: Form Length

*RUNOPT
> If a query definition is being used when the query is run, the form size specified in the query definition is used. If the form size specified in the query definition is blank, the value from QPQUPRFIL is assumed.

form-length
> Specify the form length used when this query is run. Valid values range from 1 through 255.

Element 2: Form Width

*RUNOPT
> If a query definition is being used when the query is run, the form size specified in the query definition is used. If the form size specified in the query definition is blank, the value from QPQUPRFIL is assumed.

form-width
> Specify the form width used when this query is run. Valid values range from 1 through 378.

Form type (FORMTYPE Parameter)

Specifies the type of form on which the output is printed. The identifiers used to indicate the type of forms are user-defined and can be a maximum of 10 characters in length.

Note: If a value is not specified in the query or on this parameter, or if a query name is not specified, the value in QPQUPRFIL is assumed. If an override is in effect for the printer file QPQUPRFIL, this parameter uses the value specified by the override.

The possible values are:

*RUNOPT
> If a query definition is being used, the form type specified in the query definition is used when this query is run.

form-type
> Specify the form type that is used when this query is run.

Copies (COPIES Parameter)

Specifies the number of copies being printed.

Note: If a value is not specified in the query or on this parameter, or if a query name is not specified, 1 is the assumed number of copies. If an override is in effect for the printer file QPQUPRFIL, this parameter uses the value specified by the override.

The possible values are:

*RUNOPT
> If a query definition is being used, the number of copies specified in the query definition is used when this query is run. If the number of copies specified in the query definition is blank, the number of copies from QPQUPRFIL is assumed.

number-of-copies
> Specify the number of copies to print when this query is run. Specify a number ranging from 1 through 255.

Line spacing (LINESPACE Parameter)

Specifies the number of blank lines to leave between lines in the report. The numbers range from 1 through 3. If a value was not specified in the query or on this parameter, or if a query name is not specified, 1 is the assumed value.

The possible values are:

*RUNOPT
> If a query definition is being used, the number of lines specified in the query definition is used when this query is run.

1   Indicates that single spacing (no blank lines) is used when the query output is printed.

2   Indicates that double spacing (1 blank line) is used when the query output is printed.

3   Indicates that triple spacing (2 blank lines) is used when the query output is printed.

Output file (OUTFILE Parameter)

Specifies the database file (if any) that receives the query output. If no value is specified for this parameter, the library, file, member, and option specified in the query are assumed. If a query is not specified, the file QQRYOUT is created in the default output file library. This library is defined in the query profile or the current library (*CURLIB) if no profile exists. The first member (*FIRST) of this new file is used for the output.

Note: If you did not specify a current library, the QGPL library is used.

If the name specified by the OUTFILE parameter does not exist, the system creates it in the specified library.

There are three parts to this parameter.

Element 1: Database File Name The possible values are:

data-base-file-name
> Specify the name of the database file that receives the output of the query.

*RUNOPT

> The database file specified in the query is used to receive the output of the query. The file, library, member, and option do not change. Any values specified for the library, member, and option are ignored.

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

*RUNOPT

> If specified in the query definition, the output is directed to the library named in the query definition.

*CURLIB

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

library-name

> Specify the name of the library to be used.

Element 2: Database File Member

The possible member values are:

*FIRST

> The first member in the file is used to receive the query output.

*LAST

> The last member in the file is used to receive the query output.

*RUNOPT

> The member specified in the query is used to receive the query output.

*ALL   The output file is a partitioned table, where all members in the file are used to receive the query output. When *ALL is specified for the member, the **Element 3 Data** option can only be set to *RPLMBR or *ADDMBR. The partitoned table must already exist when the query is run.

member-name

> The named file member is used to receive the query output.

Element 3: Adding or Replacing Data

The last option specifies whether to put the data in a new database file, replace an existing database file, add a new member, replace an existing member, or add data to an existing member. If no value is specified in the query or in this parameter, or if a query name is not specified, the value *NEWFILE is assumed.

*RUNOPT

> If a query definition is used, the member option specified in the query definition is the type used when this query is run.

*NEWFILE

> The output is written to a new database file. This option is not valid when the member name is set to *ALL.

*RPLFILE

> The output deletes the old file and creates a new file. This option is not valid when the member name is set to *ALL.

*NEWMBR

> The output is added as a new member. This option is not valid when the member name is set to *ALL.

*RPLMBR
>   The existing member is cleared and the output is then added.

*ADDMBR
>   The output is added to the end of an existing member.

Authority (AUT Parameter)

Specifies the authority given to users who do not have specific authority to the output file, who are not on an authorization list, and whose user group has no specific authority to the output file. More information on this parameter is in the CL Reference book, Appendix A.

Note: The authority is assigned only when the output file is created or replaced.

The possible values are:

*RUNOPT
>   If specified in the query definition, the authority named in the query definition is used.

*LIBCRTAUT
>   The public authority for the output file is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the output file). The public authority is determined when the output file is created. If the CRTAUT value for the library changes after the output file is created, the new value does not affect any existing objects.

*CHANGE
>   You can perform all operations on the output file except those limited to the owner or controlled by object existence authority and object management authority. You can change and perform basic functions on the output file. Change authority provides object operational authority and all data authority.

*USE   You can perform basic operations on the output file, such as running a program or reading a file. You cannot change the output file. *USE authority provides object operational authority, read authority, and execute authority.

*ALL   You perform all operations on the output file except those limited to the owner or controlled by authorization list management authority.

*EXCLUDE
>   You cannot access the output file.

authorization-list-name
>   Specify the name of the authorization list used.

Examples

Example 1: Printing Summary Records Only

```
RUNQRY   QRY(LIBX/QRY1)  OUTTYPE(*PRINTER)
         OUTFORM(*SUMMARY)  COPIES(4)
```

This command runs the query QRY1 located in library LIBX. The report that is produced and printed contains summary records only. Four copies of the report are printed.

Example 2: Running a Default Query

```
RUNQRY   QRYFILE((LIBX/FILE2 *FIRST))
         OUTTYPE(*OUTFILE)  OUTFORM(*DETAIL)
         RCDSLT(*YES)  OUTFILE(LIB2/OUT1 MBR4 *NEWMBR)
```

This command runs a default query and gets the data from the first member of file FILE2 located in library LIBX. Member MBR4 is created as a new member to file OUT1 in library LIB2. Member MBR4 contains the output from the default query. The record selection display is shown to allow you to specify which records from file FILE2 in library LIBX are written to new member MBR4 in file OUT1 in library LIB2. The output contains detail records only.

Top

Error messages

None

Top

Run Remote Command (RUNRMTCMD)

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

The Run Remote Command (RUNRMTCMD) command, also known as AREXEC when an SNA address is specified for the remote location name, allows server 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 an rexecd (remote executing) daemon if you specify *IP for the address type, or an AREXECD (APPC remote executing) daemon if you specify *SNA for the address type.

When the command is sent to the remote system, the local system waits for the command to complete and the output from the remote command will be placed in a spooled file.

Parameters

Keyword	Description	Choices	Notes
CMD	Command	*Character value*	Required, Positional 1
RMTLOCNAME	Remote location	*Element list*	Required, Positional 2
	Element 1: Name or address	*Character value*	
	Element 2: Type	***SNA**, *IP	
RMTUSER	Remote user ID	*Character value*, ***NONE**, *CURRENT	Optional
RMTPWD	Remote password	*Character value*, ***NONE**	Optional
MODE	Mode	*Communications name*, ***NETATR**	Optional
CCSID	Coded character set ID	1-65533, ***CALC**	Optional
WAITTIME	Wait time (in seconds)	2-3600, ***NOMAX**, *NOWAIT	Optional

Command (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.

Remote location (RMTLOCNAME)

Specifies the name of the target system and the type of connection used to connect to the target system.

The possible **name or address** values are dependent on the address type.

When the address type is *SNA, specify the name of the target system using the format nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name. If no network ID is specified, the network attributes are used to determine the default network ID.

When the address type is *IP, specify a host name or an internet address to identify the target system. Host names must follow these conventions:

- The first character must be either A through Z or 0 through 9.
- The last character must be either A through Z, 0 through 9, or a period (.).
- Blanks ( ) are not allowed.
- The special characters, period (.), underscore (_), and minus (-) are allowed.
- Parts of the name separated by periods (.) cannot exceed 63 characters in length.

    Note: Each part of the name separated by periods (.) must begin and end with an English alphabetic character or a numeric character.

- Names must be from 1 to 255 characters in length.

The 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 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.

The possible **address type** values are:

*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.

Remote user ID (RMTUSER)

Specifies the user identifier (ID) for the target system. If a user ID is specified for this parameter and password security is active on the target system, RMTPWD(*NONE) is not valid.

The possible values are:

*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.

Remote password (RMTPWD)

Specifies the password sent to the target system.

The possible values are:

*NONE
> The system does not send a password. If a user identifier (ID) is specified on the RMTUSER parameter and password security is active on the target system, the command will fail.

password
> Specify a password sent to the target system to verify the sign-on of the user ID specified in the RMTUSER parameter. If an address type of *IP is specified, the password sent is not substituted across the communication line; if *SNA is specified, the password may or may not be substituted, depending on whether the remote system supports password substitution.

Mode (MODE)

Specifies the name of the mode to be used when sending the command to the target system. This parameter is only allowed when the address type value is *SNA.

The possible values are:

*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.

Coded character set ID (CCSID)

Specifies the coded character set identifier (CCSID) that will be used for conversions. When data is sent to the remote system, the data is converted using the CCSID of the job to the CCSID specified on the CCSID parameter. Similarly, when data is received from the remote system, the data is converted from the CCSID that was specified on the CCSID parameter to the CCSID of the job.

Restrictions:
1. If the CCSID of the job is 65535 (indicating no conversion), the default CCSID of the job will be used.
2. This command uses the round-trip conversion method when converting data from the source CCSID to the target CCSID and back. For more information about CCSID conversion methods, see the National Language Support book.

The possible values are:

*CALC
> The CCSID value sent from the target system is used if it is available. If a value is not provided from the target system, a default value of 00819 (ISO 8859-1 8-bit ASCII) will be used.

coded-character-set-identifier
> The requested CCSID value is used. This value is validated to ensure a valid CCSID has been requested.

Wait time (in seconds) (WAITTIME)

Specifies the time in seconds to wait for the return (echo) before declaring the remote location to be unreachable.

The possible values are:

*NOMAX
> The system waits forever.

*NOWAIT
> The system returns immediately if there is a connection ready and available.

time-to-wait-for-reply
> Specify the length of time in seconds. Valid values range from 2 through 3600 seconds.

Examples

Example 1: Run Remote Command (RUNRMTCMD) Using SNA Address Type

```
RUNRMTCMD   CMD('dir')  RMTLOCNAME((APPN.RMTSYS *SNA))
```

This command runs the 'dir' command on the system specified by 'RMTLOCNAME' using an SNA address type.

Example 2: Run Remote Command (AREXEC) Using SNA Address Type

```
AREXEC   CMD('dir')   RMTLOCNAME(APPN.RMTSYS)
```

This command is equivalent to the command in example 1.

Example 3: Run Remote Command Using IP Address Type and Host Name to Specify Target System

```
RUNRMTCMD   CMD('ls')  RMTLOCNAME((MYSYS.NET1.LOCAL *IP))
```

This command runs the 'ls' command on the system specified by host name MYSYS.NET1.LOCAL.

Example 4: Run Remote Command Using IP Address Type and Internet Address to Specify Target System

```
RUNRMTCMD   CMD('ls')  RMTLOCNAME(('9.5.1.94' *IP))
```

This command runs the 'ls' command on the system specified by internet address '9.5.1.94'.

Example 5: Run Remote Command With Multiple Commands Using IP Address Type and Internet Address to Specify Target System

```
RUNRMTCMD   CMD('ls; cat myfile; date')  RMTLOCNAME(('9.5.1.94' *IP))
```

This command runs multiple commands, first 'ls', then 'cat myfile', then 'date' on system specified by internet address '9.5.1.94'.

Example 6: Run Remote Command With a Wait Time Expiration Value

```
RUNRMTCMD  CMD('dir')  RMTLOCNAME((APPN.RMTSYS *SNA))  WAITTIME(15)
```

This command runs the 'dir' command on the specified system. The maximum wait time for the remote location to respond is 15 seconds. A wait time is only allowed when using *SNA address type value.

Error messages

*ESCAPE Messages

CPF91CB
Problems occurred on the command, but the command completed.

CPF91CC
Command did not complete successfully.

CPF91CF
Command failed on remote system.

CPF91C9
MODE value not allowed when *IP specified for RMTLOCNAME address type.

Run SQL Statements (RUNSQLSTM)

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

The Run SQL Statements (RUNSQLSTM) command processes a source file of Structure Query Language (SQL) statements.

Parameters

Keyword	Description	Choices	Notes
SRCFILE	Source file	*Qualified object name*	Required, Positional 1
	Qualifier 1: Source file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*	Required, Positional 2
COMMIT	Commitment control	***CHG**, *UR, *CS, *ALL, *RS, *NONE, *NC, *RR	Optional, Positional 3
NAMING	Naming	***SYS**, *SQL	Optional, Positional 4
ERRLVL	Severity level	0-40, **10**	Optional
DATFMT	Date format	***JOB**, *USA, *ISO, *EUR, *JIS, *MDY, *DMY, *YMD, *JUL	Optional
DATSEP	Date separator character	***JOB**, '/', '.', ',', '-', ' ', *BLANK	Optional
TIMFMT	Time format	***HMS**, *USA, *ISO, *EUR, *JIS	Optional
TIMSEP	Time separator character	***JOB**, ':', '.', ',', ' ', *BLANK	Optional
DFTRDBCOL	Default collection	*Name*, ***NONE**	Optional
SAAFLAG	IBM SQL flagging	***NOFLAG**, *FLAG	Optional
FLAGSTD	ANS flagging	***NONE**, *ANS	Optional
DECMPT	Decimal Point	***JOB**, *SYSVAL, *PERIOD, *COMMA	Optional
SRTSEQ	Sort sequence	Single values: ***JOB**, *LANGIDUNQ, *LANGIDSHR, *HEX Other values: *Qualified object name*	Optional
	Qualifier 1: Sort sequence	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LANGID	Language id	*Character value*, ***JOB**	Optional
PRTFILE	Print file	*Qualified object name*	Optional
	Qualifier 1: Print file	*Name*, **QSYSPRT**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
PROCESS	Statement processing	***RUN**, *SYN	Optional
ALWCPYDTA	Allow copy of data	***OPTIMIZE**, *YES, *NO	Optional
ALWBLK	Allow blocking	***ALLREAD**, *NONE, *READ	Optional
SQLCURRULE	SQL rules	***DB2**, *STD	Optional

Keyword	Description	Choices	Notes
DECRESULT	Decimal result options	*Element list*	Optional
	Element 1: Maximum precision	**31**, 63	
	Element 2: Maximum scale	0-63, **31**	
	Element 3: Minimum divide scale	0-9, **0**	
OUTPUT	Listing output	***NONE**, *PRINT	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**	Optional
DBGVIEW	Debugging view	***NONE**, *SOURCE, *STMT, *LIST	Optional
CLOSQLCSR	Close SQL cursor	***ENDACTGRP**, *ENDMOD	Optional
DLYPRP	Delay PREPARE	***NO**, *YES	Optional
USRPRF	User profile	***NAMING**, *USER, *OWNER	Optional
DYNUSRPRF	Dynamic user profile	***USER**, *OWNER	Optional

Source file (SRCFILE)

Specifies the source file that contains the Structured Query Language (SQL) statements to be run.

Qualifier 1: Run SQL Statements

name   Specify the name of the source file that contains the SQL statements to be run. The source file can be a database file or an inline data file.

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Source member (SRCMBR)

Specifies the source file member that contains the Structured Query Language (SQL) statements to be run.

name   Specify the name of the source file member that contains the SQL statements to be run.

Commitment control (COMMIT)

Specifies whether SQL statements are run under commitment control.

***CHG** or ***UR**
        Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

*CS Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

*ALL *or* ***RS**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

*NONE *or* ***NC**
Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP SCHEMA statement is included in the program, *NONE or *NC must be used.

*RR Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

Naming convention (NAMING)

Specifies the naming convention used for objects in SQL statements.

*SYS The system naming convention (library-name/file-name) is used.

*SQL The SQL naming convention (schema-name.table-name) is used.

Severity level (ERRLVL)

Specifies whether the processing is successful, based on the severity of the messages generated by the processing of the SQL statements. If errors that are greater than the value specified for this parameter occur during processing, no more statements are processed and the statements are rolled back if they are running under commitment control.

10 Statement processing is stopped when error messages with a severity level greater than 10 are received.

0-40 Specify the severity level to be used.

Date format (DATFMT)

Specifies the format used when accessing date result columns. For input date strings, the specified value is used to determine whether the date is specified in a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

*JOB The format specified for the job is used. Use the Display Job (DSPJOB) command to determine the current date format for the job.

*USA The United States date format **mm/dd/yyyy** is used.

*ISO The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

*EUR The European date format **dd.mm.yyyy** is used.

*JIS The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

*MDY The date format **mm/dd/yy** is used.

*DMY The date format **dd/mm/yy** is used.

*YMD The date format **yy/mm/dd** is used.

*JUL The Julian date format **yy/ddd** is used.

<div align="right">Top</div>

Date separator character (DATSEP)

Specifies the separator used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified for the **Date format (DATFMT)** parameter.

*JOB The date separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current date separator value for the job.

'/' A slash is used as the date separator.

'.' A period is used as the date separator.

'-' A dash is used as the date separator.

',' A comma is used as the date separator.

' ' or *BLANK
A blank is used as the date separator.

<div align="right">Top</div>

Time format (TIMFMT)

Specifies the format used when accessing time result columns. For input time strings, the specified value is used to determine whether the time is specified in a valid format.

Note: An input time string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

*HMS The **hh:mm:ss** format is used.

*USA The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

*ISO The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

*EUR The European time format **hh.mm.ss** is used.

*JIS The Japanese Industrial Standard time format **hh:mm:ss** is used.

<div align="right">Top</div>

Time separator character (TIMSEP)

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when *HMS is specified for the **Time format (TIMFMT)** parameter.

*JOB   The time separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

        Use the Display Job (DSPJOB) command to determine the current time separator value for the job.

':'   A colon is used as the time separator.

'.'   A period is used as the time separator.

','   A comma is used as the time separator.

' ' or *BLANK
        A blank is used as the time separator.

<div align="right">Top</div>

Default collection (DFTRDBCOL)

Specifies the name of the schema identifier used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers. This parameter applies only to static SQL statements.

*NONE
        The naming convention specified for the **Naming convention (NAMING)** parameter is used.

name   Specify the name of the schema identifier to be used instead of the naming convention specified for the NAMING parameter.

<div align="right">Top</div>

IBM SQL flagging (SAAFLAG)

Specifies the IBM SQL flagging function. This parameter allows you to flag SQL statements to verify whether they conform to IBM SQL syntax.

*NOFLAG
        No checks are made to see whether SQL statements conform to IBM SQL syntax.

*FLAG
        Checks are made to see whether SQL statements conform to IBM SQL syntax.

<div align="right">Top</div>

ANS flagging (FLAGSTD)

Specifies whether non-standard statements are flagged. This parameter allows you to flag SQL statements to verify whether they conform to the Core level of the ISO/IEC 9075-2003 standards.

*NONE
        No checks are made to see whether SQL statements conform to ANSI standards.

*ANS   Checks are made to see whether SQL statements conform to standards.

Decimal Point (DECMPT)

Specifies the decimal point value used for numeric constants in SQL statements. This value is also used as the decimal point character when casting between character and numeric values.

*JOB The representation for the decimal point is the value used by the job running the statement.

*SYSVAL
   The QDECFMT system value is used as the decimal point.

*PERIOD
   A period represents the decimal point.

*COMMA
   A comma represents the decimal point.

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons in SQL statements.

Single values

*JOB The SRTSEQ value for the job is used.

*LANGIDUNQ
   The unique-weight sort table for the language specified for the **Language id (LANGID)** parameter is used.

*LANGIDSHR
   The shared-weight sort table for the language specified for the LANGID parameter is used.

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

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with this program.

Qualifier 2: Library

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

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

name Specify the name of the library to be searched.

Language id (LANGID)

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

*JOB The LANGID value for the job is retrieved.

language-identifier
>   Specify a language identifier.

Print file (PRTFILE)

Specifies the printer device file to which the RUNSQLSTM printout is directed. The file must have a minimum length of 132 bytes. If a file with a record length of less than 132 bytes is specified, information is lost.

Qualifier 1: Print file

QSYSPRT
>   The RUNSQLSTM output file is directed to the IBM-supplied printer file, QSYSPRT.

name   Specify the name of the printer device file to which the RUNSQLSTM output is directed.

Qualifier 2: Library

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

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

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

Statement processing (PROCESS)

Specifies whether SQL statements in the source file member are executed or syntax-checked only.

*RUN   Statements are syntax-checked and run.

*SYN   Statements are syntax-checked only.

Allow copy of data (ALWCPYDTA)

Specifies whether a copy of the data can be used in a SELECT statement.

*OPTIMIZE
>   The system determines whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which method provides the best performance. If the **Commitment control (COMMIT)** parameter is not *NONE, the **Allow blocking (ALWBLK)** parameter should be set to *ALLREAD, when possible, for best performance.

*YES   A copy of the data is used only when necessary.

*NO   A copy of the data is not used. If a temporary copy of the data is required to perform the query, an error message is returned.

Allow blocking (ALWBLK)

Specifies whether the database manager can use record blocking and the extent to which blocking can be used for read-only cursors.

*ALLREAD

Rows are blocked for read-only cursors. All cursors in a program that are not explicitly able to be changed are opened for read-only processing even though there may be EXECUTE or EXECUTE IMMEDIATE statements in the program.

Specifying *ALLREAD:

- Allows record blocking for all read-only cursors.
- Can improve the performance of almost all read-only cursors in programs, but limits queries in the following ways:
  - The Rollback (ROLLBACK) command, a ROLLBACK statement in host languages, or the ROLLBACK HOLD SQL statement does not reposition a read-only cursor when *ALLREAD is specified.
  - Dynamic running of a positioned UPDATE or DELETE statement (for example, using EXECUTE IMMEDIATE), can not be used to update a row in a cursor unless the DECLARE statement for the cursor includes the FOR UPDATE clause.

*NONE

Rows are not blocked for retrieval of data for cursors.

Specifying *NONE:

- Guarantees that the data retrieved is current.
- May reduce the amount of time required to retrieve the first row of data for a query.
- Stops the database manager from retrieving a block of data rows that is not used by the program when only the first few rows of a query are retrieved before the query is closed.
- Can degrade the overall performance of a query that retrieves a large number of rows.

*READ

Records are blocked for read-only retrieval of data for cursors when:

- *NONE is specified for the **Commitment control (COMMIT)** parameter, which indicates that commitment control is not used.
- The cursor is declared with a FOR READ ONLY clause or there are no dynamic statements that could run a positioned UPDATE or DELETE statement for the cursor.

<div align="right">Top</div>

SQL rules (SQLCURRULE)

Specifies the semantics used for SQL statements.

*DB2 The semantics of all SQL statements will default to the rules established for DB2. The following semantics are controlled by this option:

Hexadecimal constants are treated as character data.

*STD The semantics of all SQL statements will default to the rules established by the ISO and ANSI SQL standards. The following semantics are controlled by this option:

Hexadecimal constants are treated as binary data.

<div align="right">Top</div>

Decimal result options (DECRESULT)

Specifies the maximum precision, maximum scale and minimum divide scale that should be returned for result data types. The specified limit only applies to numeric (zoned) and decimal (packed) data types used in arithmetic expressions and in SQL column functions AVG and SUM.

Element 1: Maximum precision

31   The maximum precision (length) that should be returned for the result data types is 31 digits.

63   The maximum precision (length) that should be returned for the result data types is 63 digits.

Element 2: Maximum scale

31   The maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types is 31 digits.

0-63   Specify the maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types. The value can range from 0 to the maximum precision.

Element 3: Minimum divide scale

0   The minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types is 0.

0-9   Specify the minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types. The value cannot exceed the maximum scale. If 0 is specified for the maximum scale, minimum divide scale is not used.

<div align="right">Top</div>

Listing output (OUTPUT)

Specifies whether the precompiler listing is generated.

Note: This parameter applies only to CREATE statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

*NONE
      The precompiler listing is not generated.

*PRINT
      The precompiler listing is generated.

<div align="right">Top</div>

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object being created.

Note: This parameter applies only to CREATE statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release.

*CURRENT

The object is to be used on the release of the operating system currently running on your system. The object can also be used on a system with any subsequent release of the operating system installed.

target-release

Specify the release in the format VxRxMx. The object can be used on a system with the specified release or with any subsequent release of the operating system installed.

Debugging view (DBGVIEW)

Specifies the type of source debug information to be provided by SQL.

Note: This parameter applies only to CREATE statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

*NONE

No debug view information is generated.

*SOURCE

Generates a source view of the C source generated by RUNSQLSTM for the SQL procedures, functions, or triggers in the input SQL source member. The C source member is passed to the SQL precompiler by invoking the CRTSQLCI (Create SQL ILE C object) command. A source view is also generated by the SQL precompiler for the C source member which is produced by the precompiler.

*STMT

Allows the compiled object to be debugged using program statement numbers and symbolic identifiers.

*LIST   Generates the listing view for debugging the compiled object.

Close SQL cursor (CLOSQLCSR)

Specifies when SQL cursors are implicitly closed, SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released. SQL cursors are explicitly closed when the user issues the CLOSE, COMMIT, or ROLLBACK (without HOLD) SQL statements.

Note: This parameter applies only to CREATE statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

*ENDACTGRP

SQL cursors are closed and SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released when the activation group ends.

*ENDMOD

SQL cursors are closed and SQL prepared statements are implicitly discarded when the module is exited. LOCK TABLE locks are released when the first SQL program on the call stack ends.

Delay PREPARE (DLYPRP)

Specifies whether the dynamic statement validation for a PREPARE statement is delayed until an OPEN, EXECUTE, or DESCRIBE statement is run. Delaying validation improves performance by eliminating redundant validation.

Note: This parameter applies only to CREATE statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

*NO Dynamic statement validation is not delayed. When the dynamic statement is prepared, the access plan is validated. When the dynamic statement is used in an OPEN or EXECUTE statement, the access plan is revalidated. Because the authority or the existence of objects referred to by the dynamic statement may change, you must still check the SQLCODE or SQLSTATE after issuing the OPEN or EXECUTE statement to ensure that the dynamic statement is still valid.

*YES Dynamic statement validation is delayed until the dynamic statement is used in an OPEN, EXECUTE, or DESCRIBE SQL statement. When the dynamic statement is used, the validation is completed and an access plan is built. If you specify *YES on this parameter for precompiled programs, you should check the SQLCODE and SQLSTATE after running an OPEN, EXECUTE, or DESCRIBE statement to ensure that the dynamic statement is valid.

> **Note:** If you specify *YES, performance is not improved if the INTO clause is used on the PREPARE statement or if a DESCRIBE statement uses the dynamic statement before an OPEN is issued for the statement.

User profile (USRPRF)

Specifies the user profile that is used when the compiled program object and SQL package object is run, including the authority that the program object or SQL package has for each object in static SQL statements. The profile of either the owner or the user is used to control access to objects.

Note: This parameter applies only to CREATE statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

*NAMING
The user profile is determined by the naming convention. If the naming convention is *SQL, USRPRF(*OWNER) is used. If the naming convention is *SYS, USRPRF(*USER) is used.

*USER
The profile of the user running the program or SQL package is used.

*OWNER
The user profiles of both the owner and the user are used when the program or SQL package is run.

Dynamic user profile (DYNUSRPRF)

Specifies the user profile used for dynamic SQL statements.

Note: This parameter applies only to CREATE statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

*USER

Local dynamic SQL statements are run under the profile of the program's user. Distributed dynamic SQL statements are run under the profile of the application server job.

*OWNER

Local dynamic SQL statements are run under the profile of the program's owner. Distributed dynamic SQL statements are run under the profile of the SQL package's owner.

<div align="right">Top</div>

Examples

```
RUNSQLSTM  SRCFILE(MYLIB/MYFILE)  SRCMBR(MYMBR)
```

This command processes the SQL statements in member MYMBR found in file MYFILE in library MYLIB.

<div align="right">Top</div>

Error messages

*ESCAPE Messages

SQL9006

DB2 UDB Query Manager and SQL Development Kit for iSeries not at same install level as i5/OS.

SQL9010

RUNSQLSTM command failed.

SQL9014

Remote Connection Active

<div align="right">Top</div>

Revoke Access Code Authority (RVKACCAUT)

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

The Revoke Access Code Authority (RVKACCAUT) command allows you to revoke the access code authority for an individual user or a group of users.

Restrictions:

You must have all object (*ALLOBJ) special authority to revoke access code authority for other users.

Parameters

Keyword	Description	Choices	Notes
ACC	Document access code	Single values: *ALL Other values (up to 300 repetitions): 1-2047	Required, Positional 1
USER	User profile	Single values: *CURRENT, *ALL Other values (up to 300 repetitions): *Name*	Required, Positional 2

Document access code (ACC)

Specifies the access code for which authority is revoked.

This is a required parameter.

*ALL All access code authority for the user is revoked.

1-2047 Specify the access code for which authority is revoked. A maximum of 300 access codes can be specified here.

User profile (USER)

Specifies the names of the user profile for which access code authority is revoked.

This is a required parameter.

*CURRENT
> Specifies that the access code authority of the user currently running this command is revoked. This is the only valid option you can specify unless you have all object (*ALLOBJ) or security administrator (*SECADM) special authority.

> **Note:** You can specify *CURRENT on the USER parameter to remove authority from any access codes to which you have authority.

*ALL Access code authority is revoked from all users.

313

name   Specify the name of the user profile for which access code authority is revoked.

Examples

Example 1: Revoking Authority of Current User

```
RVKACCAUT   ACC(250)  USER(*CURRENT)
```

This command takes away the access code authority of access code 250 from the user currently running this command.

Example 2: Revoking Authority of Specific User

```
RVKACCAUT   ACC(300)  USER(BILLY)
```

This command takes away the access code authority of access code 300 from user BILLY. This command must be run by someone with *ALLOBJ or *SECADM special authority, or by user BILLY. A user who runs this command for himself can enter USER(*CURRENT) or his own user profile name; they are the same.

Error messages

*ESCAPE Messages

CPF9009

System requires file &1 in &2 be journaled.

CPF9014

Access code authority removed from &1 users, not revoked from &2 users.

CPF9024

System cannot get correct record to finish operation.

CPF9037

Not allowed to specify USER(*ALL).

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

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

Revoke Object Authority (RVKOBJAUT)

The Revoke Object Authority (RVKOBJAUT) command is used to take away specific (or all) authority for the named object(s) from one or more users named in the command, or to remove the authority of an authorization list for the named object(s). This command can be run by the security officer, by an object's owner, or by a user who has object management authority for the object to be revoked. Users who have object management authority can revoke only the explicit authority that they have. A user may not be able to grant or revoke authority for an object that has been allocated (locked) by another job. Authority cannot be revoked for an object that is currently in use.

Note: Caution should be used when changing the public authority on IBM-supplied objects. For example, changing the public authority on the QSYSOPR message queue to be more restrictive than *CHANGE will cause some system programs to fail. The system programs will not have enough authority to send messages to the QSYSOPR message queue. For more information, refer to the iSeries Security Reference, SC41-5302.

Restrictions:
- Before this command is used to remove authorities to use a device, control unit, or line description, its associated device, control unit, or line must be varied on.
- Authority to use a device cannot be revoked if a user is currently signed on to the device.

  Note: Users can revoke their own authority to a device if they are currently signed onto that device. However, doing so may produce unpredictable results and is not advisable.
- For display stations or for work station message queues associated with the display station, if this command is not run from the device for which authorities are to be revoked, it should be preceded by the Allocate Object (ALCOBJ) command and followed by the Deallocate Object (DLCOBJ) command.
- Object type *DOC or *FLR cannot be specified.
- Document interchange support must be used.
- Object type *AUTL cannot be specified. The Change Authorization List Entry (CHGAUTLE) or Remove Authorization List Entry (RMVAUTLE) commands must be used. AUT (*AUTL) can be specified only with USER (*PUBLIC).
- Only a user with *ALL authority or the owner can remove the authorization list.
- You must have *USE authority to the auxiliary storage pool device if one is specified.

*** Security Risk ***

Revoking all authorities specifically given to a user for an object can result in the user having more authority than before the revoke operation. If a user has *USE authority for an object and *CHANGE authority on the authorization list that secures the object, revoking *USE authority results in the user having *CHANGE authority to the object.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	*Qualified object name*	Required, Positional 1
	Qualifier 1: Object	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB, *ALL, *ALLUSR, *USRLIBL, *ALLAVL, *ALLUSRAVL	
OBJTYPE	Object type	*ALL, *ALRTBL, *BNDDIR, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CNNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DTAARA, *DTADCT, *DTAQ, *EDTD, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *IPXD, *JOBD, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *LOCALE, *M36, *M36CFG, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *NTBD, *NWID, *NWSCFG, *NWSD, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMQRY, *QRYDFN, *RCT, *S36, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *TBL, *TIMZON, *USRIDX, *USRPRF, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 2
ASPDEV	ASP device	*Name*, ******, *SYSBAS	Optional
USER	Users	Single values: *ALL, *PUBLIC Other values (up to 50 repetitions): *Name*	Optional, Positional 3
AUT	Authority	Single values: ***CHANGE**, *ALL, *USE, *EXCLUDE, *AUTL Other values (up to 10 repetitions): *OBJALTER, *OBJEXIST, *OBJMGT, *OBJOPR, *OBJREF, *ADD, *DLT, *READ, *UPD, *EXECUTE	Optional, Positional 4
AUTL	Authorization list	*Name*	Optional

Object (OBJ)

Specifies the objects to have specific authority revoked. If *ALL is specified for the object name, a library name must be specified.

This is a required parameter.

Qualifier 1: Object

*ALL   All objects of the specified type (OBJTYPE) found in the search have specific authorities revoked. You must specify the name of a library when *ALL is specified for the object name.

generic-name
            Specify the generic name of the objects that are to have specific authorities revoked.

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

name   Specify the name of the object that is to have specific authorities revoked.

Qualifier 2: Library

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

*CURLIB
>   The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*USRLIBL
>   If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*ALL   All the libraries in the auxiliary storage pools (ASPs) specified for the **ASP device (ASPDEV)** parameter are searched.

*ALLUSR
>   All user libraries in the auxiliary storage pools (ASPs) defined by the **ASP device (ASPDEV)** parameter are searched.

>   User libraries are all libraries with names that do not begin with the letter Q except for the following:

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

>   Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

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

>   1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).

>   2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

*ALLAVL
>   All libraries in all available ASPs are searched.

*ALLUSRAVL
>   All user libraries in all available ASPs are searched. Refer to *ALLUSR for a definition of user libraries.

name   Specify the name of the library to be searched.

Object type (OBJTYPE)

Specifies the object type of the object that has specific authorities revoked. For a complete list of object types, position the cursor on this parameter while prompting the command and press F4.

This is a required parameter.

*ALL   All object types (except *AUTL) have specific authorities revoked.

object-type
>   Specify the object type that is to have specific authorities revoked.

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device name where the library that contains the object (OBJ parameter) is located. If the object's library resides in an ASP that is not part of the library name space associated with the job, this parameter must be specified to ensure the correct object is used as the target of this command's operation.

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

*SYSBAS
>   The system ASP and all basic user ASPs will be searched to locate the object. No independent ASPs will be searched, even if the job has an ASP group.

name   Specify the device name of the independent ASP to be searched to locate the object. The independent ASP must have been activated (by varying on the ASP device) and have a status of AVAILABLE. The system ASP and basic user ASPs will not be searched.

Users (USER)

Specifies one or more users whose specific authorities to the named object are to be revoked.

Note: Either this parameter or the **Authorization list (AUTL)** parameter must be specified.

Authorities revoked by this command are related to those given by the Grant Object Authority (GRTOBJAUT) command. If users have public authority to an object because USER(*PUBLIC) was specified on the GRTOBJAUT command, that public authority is revoked when *PUBLIC is specified on this parameter. If users have specific authorities to an object because their names were specified on the GRTOBJAUT command, their names must be specified on this parameter to revoke the same authorities.

The authorities to be revoked are specified on the **Authority (AUT)** parameter.

Single values

*ALL   The authorities specified are to be taken away from all enrolled users of the system except the owner, whether they were publicly or explicitly authorized.

*PUBLIC
>   The specified authorities are taken away from users who do not have specific authority for the object, are not on the authorization list, and whose group has no authority. Any users who have specific authorities still keep their authorities to the object.

Other values (up to 50 repetitions)

name Specify the name of the user profile of the user that is to have the specified authorities revoked. This parameter cannot be used to revoke public authority from specific users; only authorities that were specifically given to a user can be specifically revoked. A maximum of 50 user profile names can be specified.

Authority (AUT)

Specifies the authorities to be revoked from the users who do not have specific authority to the object, who are not on an authorization list, and whose user group does not have specific authority to the object.

Single values

*CHANGE
The user can perform all operations on the object except those limited to the owner or controlled by object existence (*OBJEXIST) and object management (*OBJMGT) authorities. The user can change and perform basic functions on the object. *CHANGE authority provides object operational (*OBJOPR) authority and all data authority. If the object is an authorization list, the user cannot add, change, or remove users.

*ALL The user can perform all operations except those limited to the owner or controlled by authorization list management (*AUTLMGT) authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the object.

*USE The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. Use (*USE) authority provides object operational (*OBJOPR), read (*READ), and execute (*EXECUTE) authorities.

*EXCLUDE
The user cannot access the workstation object.

*AUTL
The public authority of the authorization list specified on the AUTL parameter is used for the public authority for the object.

Note: You can specify AUT(*AUTL) only when USER(*PUBLIC) is also specified.

Other values (up to 10 repetitions)

*OBJALTER
Object alter authority provides the authority needed to alter the attributes of an object. If the user has this authority on a database file, the user can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user has this authority on an SQL package, the user can change the attributes of the SQL package. This authority is currently only used for database files and SQL packages.

*OBJMGT
Object management authority provides the authority to The security for the object, move or rename the object, and add members to database files.

*OBJEXIST
Object existence authority provides the authority to control the object's existence and ownership. If a user has special save system authority (*SAVSYS), object existence authority is not needed to perform save restore operations on the object.

*OBJOPR

Object operational authority provides authority to look at the description of an object and use the object as determined by the data authority that the user has to the object.

*OBJREF

Object reference authority provides the authority needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user has this authority on a physical file, the user can add referential constraints in which the physical file is the parent. This authority is currently only used for database files.

Data authorities

*ADD Add authority provides the authority to add entries to an object (for example, job entries to an queue or records to a file).

*DLT Delete authority provides the authority to remove entries from an object.

*EXECUTE

Execute authority provides the authority needed to run a program or locate an object in a library.

*READ

Read authority provides the authority needed to get the contents of an entry in an object or to run a program.

*UPD Update authority provides the authority to change the entries in an object.

<div align="right">Top</div>

Authorization list (AUTL)

Specifies the authorization list that is revoked from the object specified for the **Object (OBJ)** parameter. If public authority in the object is *AUTL, it is changed to *EXCLUDE.

Note: Either this parameter or the **Users (USER)** parameter must be specified. If this parameter is specified, the AUT parameter is ignored.

name Specify the name of the authorization list.

<div align="right">Top</div>

Examples

Example 1: Removing Authority From All Users Except Program Owner

```
RVKOBJAUT   OBJ(ARLIB/PROG1)   OBJTYPE(*PGM)   USER(*ALL)
```

This command removes the authorities (AUT was not specified; *CHANGE is assumed) from all users who were either explicitly or publicly authorized, except the owner, for the program (*PGM) named PROG1 located in the library named ARLIB.

Example 2: Removing Object Owner's Authority to Delete a Program

```
RVKOBJAUT   OBJ(TSMITHPGM/MITHLIB)   OBJTYPE(*PGM)
            USER(TMSMITH)   AUT(*OBJEXIST)
```

This command removes the object owner's (TMSMITH) 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/ARJOBD)  OBJTYPE(*JOBD)  USER(RLJOHNSON)
            AUT(*OBJEXIST)
```

This command removes the object existence authority for the object named ARJOBD from the user RLJOHNSON. ARJOBD 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

*ESCAPE Messages

CPF22A0

> Authority of *AUTL is allowed only with USER(*PUBLIC).

CPF22A1

> OBJTYPE(*AUTL) not valid on this command.

CPF22A2

> Authority of *AUTL not allowed for object type *USRPRF.

CPF22A3

> AUTL parameter not allowed for object type *USRPRF.

CPF22A4

> *EXCLUDE cannot be revoked from *PUBLIC.

CPF22A5

> Object &1 in &3 type *&2 not secured by authorization list &4.

CPF22DA

> Operation on file &1 in &2 not allowed.

CPF2207

> Not authorized to use object &1 in library &3 type *&2.

CPF2208

> Object &1 in library &3 type *&2 not found.

CPF2209

> Library &1 not found.

CPF2210

> Operation not allowed for object type *&1.

CPF2211

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

CPF2216

>Not authorized to use library &1.

CPF2224

>Not authorized to revoke authority for object &1 in &3 type *&2.

CPF2227

>One or more errors occurred during processing of command.

CPF2236

>AUT input value not supported.

CPF2243

>Library name &1 not allowed with OBJ(generic name) or OBJ(*ALL).

CPF2253

>No objects found for &1 in library &2.

CPF2254

>No libraries found for &1 request.

CPF2273

>Authority may not have been changed for object &1 in &3 type *&2 for user &4.

CPF2283

>Authorization list &1 does not exist.

CPF9804

>Object &2 in library &3 damaged.

*STATUS Messages

CPF2256

>Specified authority for the object not revoked from all users.

Top

Revoke Public Authority (RVKPUBAUT)

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

The Revoke Public Authority (RVKPUBAUT) limits the use of a set of IBM-supplied commands and programs by changing the public authority to *EXCLUDE. To determine what commands and programs are being restricted, issue the Retrieve CL Source (RTVCLSRC) command against the program QSECRVKP and examine the source file created by the RTVCLSRC command.

Restriction: You must have all object (*ALLOBJ) special authority to run this command.

This command can be customized by the security administrator by following the steps below:

1. Issue the Retrieve CL Source (RTVCLSRC) command against the program QSECRVKP.

2. Edit the source code produced from the RTVCLSRC command and compile the new program. Make sure that the program is given a new name, is created into a library other than QSYS, and that the *PUBLIC authority is set to *EXCLUDE.

3. Issue the Change Command (CHGCMD) against the Revoke Public Authority command and specify your new program for the **PGM** parameter. An example is listed below:

   CHGCMD CMD(QSYS/RVKPUBAUT) PGM(library_name/new_pgm_name)

   Note: If a product upgrade is done, the RVKPUBAUT command is reinstalled, or maintenance is applied to the RVKPUBAUT command, the CHGCMD will have to be issued again to customize the command.

DISCLAIMER: IBM cannot guarantee or imply reliability, serviceability, performance or function of the retrieved QSECRVKP source code and any programs. THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.

Parameters

Keyword	Description	Choices	Notes
LIB	Library	*Name*, **QSYS**	Optional

Library (LIB)

The name of the library where the IBM-supplied commands are to be found. The library value is only used for commands because they can be located in secondary language libraries.

QSYS   The commands are located in library QSYS.

library-name
    The library where the commands are located.

Examples

```
RVKPUBAUT  LIB(QSYS)
```

This command sets the public authority for commands and programs in library QSYS to *EXCLUDE.

Error messages

*ESCAPE Messages

CPFB304

   User does not have required special authorities.

Revoke User Permission (RVKUSRPMN)

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

The Revoke User Permission (RVKUSRPMN) command allows you to revoke user permission from one user (or all users) to access documents or folders on behalf of another user.

Note: If work is being done on behalf of another user at the time this command is running, functions that have started are completed; however, additional functions are not accepted.

Restrictions:

You must have all object (*ALLOBJ) special authority to revoke document authority for other users.

Parameters

Keyword	Description	Choices	Notes
FROMUSER	From user profile	*Name*, *ALL	Required, Positional 1
FORUSER	For user profile	Single values: ***CURRENT** Other values (up to 300 repetitions): *Name*	Optional, Positional 2

From user profile (FROMUSER)

Specifies the user profile name of the user whose permission is revoked.

This is a required parameter.

name   Specify the name of the user profile that is no longer permitted to work on behalf of the user specified on the **For user profile (FORUSER)** parameter.

*ALL   All users who are currently permitted to work on behalf of other users are no longer permitted to do so.

For user profile (FORUSER)

Specifies the user profile name of the user on whose behalf the user specified on the **From user profile (FROMUSER)** parameter can no longer work.

*CURRENT
        The user specified on the **From user profile (FROMUSER)** parameter can no longer work on your behalf.

name   Specify the name of the user profile on whose behalf other users are no longer permitted to work.

Examples

```
RVKUSRPMN    FROMUSER(JOHNSON)   FORUSER(ANDERSON)
```

This command takes away user permission from JOHNSON for ANDERSON. The user JOHNSON is no longer allowed to work on behalf of ANDERSON.

Error messages

*ESCAPE Messages

CPF9008

User permission revoked for &1 users, not revoked for &2.

CPF9009

System requires file &1 in &2 be journaled.

CPF9024

System cannot get correct record to finish operation.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

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

Revoke Workstation Object Aut (RVKWSOAUT)

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

The Revoke Workstation Object Authority (RVKWSOAUT) command is used to take away authority for a workstation object used by the i5/OS Graphical Operations program. Specific or all authority can be taken away from one or more users named in the command. You also can take away the authority of an authorization list for the named object.

This command can be issued by a security officer, by an object owner, or by a user who has object management authority to the object for which authority is to be revoked. If a specific authority (other than *ALL) is specified on the AUT parameter, and that authority is not revoked, a message is issued that indicates the authority that is not revoked.

*** Security Risk ***

Revoking all authorities given specifically to a user for an object can result in the user having more authority than before the operation. If a user has *USE authority for an object and *CHANGE authority on the authorization list that secures the object, revoking *USE authority results in the user having *CHANGE authority to the object.

Restrictions:

- If you have object management (*OBJMGT) authority, you can revoke only the explicit authority that you have.
- You might not be able to grant or revoke authority for an object that has been allocated (locked) to another job. Authority cannot be revoked for an object that is currently in use.

Parameters

Keyword	Description	Choices	Notes
WSOTYPE	Workstation object type	*Element list*	Required, Positional 1
	Element 1:	*TPLWRKARA, *WRKARA, *TPLPRTOL, *PRTOL, *TPLPRTL, *PRTL, *TPLOUTQ, *TPLOUTQL, *OUTQL, *TPLJOBL, *JOBL, *TPLJOBQ, *TPLJOBLOG, *JOBLOG, *TPLJOBQL, *JOBQL, *TPLMSGL, *MSGL, *TPLMSGQ, *TPLMSGSND, *MSGSND, *TPLSGNUSL, *SGNUSL, *TPLOBJL, *OBJL, *TPLLIBSL, *LIBSL, *TPLLIB, *LAUNCH, *TPLLAUNCH, *PRSSET	
USER	Users	Single values: *ALL, *PUBLIC Other values (up to 50 repetitions): *Qualifier list*	Optional, Positional 2
	Qualifier 1: Users	*Name*	
AUT	Authority	Single values: ***CHANGE**, *ALL, *USE, *EXCLUDE, *AUTL Other values (up to 7 repetitions): *OBJEXIST, *OBJMGT, *OBJOPR, *ADD, *DLT, *READ, *UPD	Optional, Positional 3
AUTL	Authorization list	*Name*	Optional

Workstation object type (WSOTYPE)

Specifies the workstation object for which specific authorities are to be revoked.

This is a required parameter.

*TPLWRKARA
> The authorities to the work area template are revoked.

*WRKARA
> The authorities to the work area objects are revoked.

*TPLPRTOL
> The authorities to the printer output list template are revoked.

*PRTOL
> The authorities to the printer output list objects are revoked.

*TPLPRTL
> The authorities to the printer list template are revoked.

*PRTL The authorities to the printer list objects are revoked.

*TPLOUTQ
> The authorities to the output queue template are revoked.

*TPLOUTQL
> The authorities to the output queue list template are revoked.

*OUTQL
> The authorities to the output queue list objects are revoked.

*TPLJOBL
> The authorities to the job list template are revoked.

*JOBL The authorities to the job list objects are revoked.

*TPLJOBQ
> The authorities to the job queue template are revoked.

*TPLJOBLOG
> The authorities to the job log template are revoked.

*JOBLOG
> The authorities to the job log objects are revoked.

*TPLJOBQL
> The authorities to the job queue list template are revoked.

*JOBQL
> The authorities to the job queue list objects are revoked.

*TPLMSGL
> The authorities to the message list template are revoked.

*MSGL
> The user authorities to the message list objects are revoked.

*TPLMSGQ
> The authorities to the message queue template are revoked.

*TPLMSGSND
> The authorities to the message sender template are revoked.

*MSGSND
>   The authorities to the message sender objects are revoked.

*TPLSGNUSL
>   The authorities to the signed-on user list template are revoked.

*SGNUSL
>   The authorities to the signed-on user list objects are revoked.

*TPLOBJL
>   The authorities to the object list template are revoked.

*OBJL   The authorities to the object list objects are revoked.

*TPLLIBSL
>   The authorities to the library list template are revoked.

*LIBSL
>   The user authorities to the library list objects are revoked.

*TPLLIB
>   The authorities to the library template are revoked.

*TPLLAUNCH
>   The authorities to the job submitter template are revoked.

*LAUNCH
>   The authorities to the job submitter objects are revoked.

*PRSSET
>   The authorities to the personal setting objects are revoked.

<div align="right">Top</div>

Users (USER)

Specifies one or more users whose specific authorities to the named object are to be revoked.

Authorities revoked by this command are related to those given by the Grant Workstation Object Authority (GRTWSOAUT) command. If users have public authority to an object because USER(*PUBLIC) was specified on the GRTWSOAUT command, that public authority is revoked when *PUBLIC is specified on this parameter. If users have specific authorities to an object because their names were specified on the GRTWSOAUT command, their names must be specified on this parameter to revoke the same authorities.

The authorities to be revoked are those specified for the **Authority (AUT)** parameter.

Note: Either this parameter or the AUTL parameter must be specified.

*ALL   The authorities specified on the AUT parameter are taken away from all enrolled users of the system except the owner, if they are publicly or explicitly authorized.

*PUBLIC
>   The specified authorities are taken away from users who do not have specific authority for the object, are not on the authorization list, and whose group has no authority. Any users who have specific authorities still keep their authorities to the object.

name   Specify the name of the user profile that is to have the specified authorities revoked. This parameter cannot be used to revoke public authority from specific users; only authorities that were specifically given to a user can be specifically revoked.

<div align="right">Top</div>

Authority (AUT)

Specifies the authority to be revoked from the users who do not have specific authority to the object, who are not on an authorization list, and whose user group does not have specific authority to the object.

Single values

*CHANGE
>The user can perform all operations on the object except those limited to the owner or controlled by object existence (*OBJEXIST) and object management (*OBJMGT) authorities. The user can change and perform basic functions on the object. *CHANGE authority provides object operational (*OBJOPR) authority and all data authority. If the object is an authorization list, the user cannot add, change, or remove users.

*ALL   The user can perform all operations except those limited to the owner or controlled by authorization list management (*AUTLMGT) authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the object.

*USE   The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. Use (*USE) authority provides object operational (*OBJOPR), read (*READ), and execute (*EXECUTE) authorities.

*EXCLUDE
>The user cannot access the workstation object.

*AUTL
>The public authority of the authorization list specified on the AUTL parameter is used for the public authority for the object.

>**Note:** You can specify AUT(*AUTL) only when USER(*PUBLIC) is also specified.

Other values (up to 10 repetitions)

*OBJALTER
>Object alter authority provides the authority needed to alter the attributes of an object. If the user has this authority on a database file, the user can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user has this authority on an SQL package, the user can change the attributes of the SQL package. This authority is currently only used for database files and SQL packages.

*OBJMGT
>Object management authority provides the authority to The security for the object, move or rename the object, and add members to database files.

*OBJEXIST
>Object existence authority provides the authority to control the object's existence and ownership. If a user has special save system authority (*SAVSYS), object existence authority is not needed to perform save restore operations on the object.

*OBJOPR
>Object operational authority provides authority to look at the description of an object and use the object as determined by the data authority that the user has to the object.

*OBJREF
>Object reference authority provides the authority needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user has this authority on a physical file, the user can add referential constraints in which the physical file is the parent. This authority is currently only used for database files.

Data authorities

*ADD Add authority provides the authority to add entries to an object (for example, job entries to an queue or records to a file).

*DLT Delete authority provides the authority to remove entries from an object.

*EXECUTE
Execute authority provides the authority needed to run a program or locate an object in a library.

*READ
Read authority provides the authority needed to get the contents of an entry in an object or to run a program.

*UPD Update authority provides the authority to change the entries in an object.

Authorization list (AUTL)

Specifies the authorization list that is revoked from the object specified on the WSOTYPE parameter. If public authority for the object is *AUTL, it is changed to *EXCLUDE. The authorization list's authority is then removed.

Note: Either this parameter or the USER parameter must be specified. If this parameter is specified, the AUT parameter is ignored.

name Specify the name of the authorization list.

Examples

RVKWSOAUT   WSOTYPE(*SGNUSL) USER(HEANDERSON) AUT(*DLT *UPD)

This command removes the delete and the update authorities for signed-on user list objects from the user profile name HEANDERSON.

Error messages

Unknown

Save Object (SAV)

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

Parameters
Examples
Error messages

The Save (SAV) command saves a copy of one or more objects that can be used in the integrated file system.

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

Restrictions:

- For detailed restrictions on using this command to save objects by using name patterns in the root directory, to save objects in libraries, or to save document library objects, see the Backup and Recovery book, SC41-5304.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Values (up to 4 repetitions): *Path name*	Required, Positional 1
OBJ	Objects	Values (up to 300 repetitions): *Element list*	Optional, Positional 2
	Element 1: Name	*Path name*, *****	
	Element 2: Include or omit	***INCLUDE**, *OMIT	
PATTERN	Name pattern	Values (up to 300 repetitions): *Element list*	Optional
	Element 1: Pattern	*Character value*, *****	
	Element 2: Include or omit	***INCLUDE**, *OMIT	
SUBTREE	Directory subtree	***ALL**, *DIR, *NONE, *OBJ, *STG	Optional
SAVACT	Save active	***NO**, *YES, *SYNC	Optional
SAVACTOPT	Save active option	***NONE**, *ALWCKPWRT	Optional
OUTPUT	Output	*Path name*, ***NONE**, *PRINT	Optional
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional
LABEL	Label	*Character value*, ***GEN**	Optional
OPTFILE	Optical file	*Path name*, *****	Optional
SEQNBR	Sequence number	1-16777215, ***END**	Optional
EXPDATE	File expiration date	*Date*, ***PERM**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
USEOPTBLK	Use optimum block	***YES**, *NO	Optional
SAVACTMSGQ	Save active message queue	*Path name*, ***NONE**, *WRKSTN	Optional
INFTYPE	Type of output information	***ALL**, *ERR, *SUMMARY	Optional
SYSTEM	System	*ALL, ***LCL**, *RMT	Optional

© Copyright IBM Corp. 1998, 2006 **333**

Keyword	Description	Choices	Notes
CHGPERIOD	Time period for last change	*Element list*	Optional
	Element 1: Start date	*Date*, ***ALL**, *LASTSAVE	
	Element 2: Start time	*Time*, ***ALL**	
	Element 3: End date	*Date*, ***ALL**	
	Element 4: End time	*Time*, ***ALL**	
PRECHK	Object pre-check	***NO**, *YES	Optional
TGTRLS	Target release	***CURRENT**, *PRV, V5R2M0, V5R3M0, V5R4M0	Optional
UPDHST	Update history	Single values: ***NO**, *YES Other values (up to 2 repetitions): *SYS, *PC	Optional
CLEAR	Clear	***NONE**, *ALL, *AFTER, *REPLACE	Optional
DTACPR	Data compression	*YES, *NO, ***DEV**, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	***DEV**, *NO	Optional
ASPDEV	ASP device	*Name*, ***DFT**, *ALLAVL, *, *SYSBAS, *CURASPGRP	Optional
SCAN	Scan objects	*Element list*	Optional
	Element 1: Scan during save	***NO**, *YES	
	Element 2: Save failed objects	***NOSAVFAILED**, *SAVFAILED	

Device (DEV)

Specifies the device to which the objects are to be saved.

For more information on specifying device path names, refer to "Specifying the device name" in the Backup and Recovery information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

This is a required parameter.

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

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

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

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

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

For information about creating a media definition, see the Create Media Definition API in the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

For information about using a media definition, see the Backup and Recovery book, SC41-5304 and the Backup Your Server topic topic in the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Objects (OBJ)

Specifies the objects to be saved. You can specify an object name pattern for the path name to be used. When a path name is specified that could match many objects, you can specify a value for the **Name pattern (PATTERN)** parameter to subset the objects that are to be saved.

A maximum of 300 path names can be specified.

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

Additional information about object name patterns is in the Integrated file system information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Element 1: Name

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

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

Element 2: Include or omit

Specifies whether names that match the pattern should be included or omitted from the operation. Note that in determining whether a name matches a pattern, relative name patterns are always treated as relative to the current working directory.

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

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

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

Name pattern (PATTERN)

Specifies one or more object name patterns to be used to subset the objects to be saved. The **Objects (OBJ)** parameter defines the set of candidate objects. A maximum of 300 values can be specified for this parameter.

Element 1: Pattern

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

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

Element 2: Include or omit

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

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

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

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

Directory subtree (SUBTREE)

Specifies whether directory subtrees are included in the save operation.

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

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

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

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

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

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state, this parameter is ignored and the save operation is performed as if SAVACT(*NO) was specified.

*NO Objects that are in use are not saved. Objects cannot be updated while being saved.

*YES Objects can be saved and used at the same time. The object checkpoints can occur at different times.

*SYNC
> Objects can be saved and used at the same time. All of the object checkpoints occur at the same time.

Save active option (SAVACTOPT)

Specifies options to be used with the save while active parameter.

*NONE

No special save while active options will be used.

*ALWCKPWRT

Enables objects to be saved while they are being updated if the corresponding system attribute
for the object is set.

Note: This option should only be used by applications to save objects that are associated with the
application and that have additional backup and recovery considerations. For more information,
refer to the Backup and Recovery information in the iSeries Information Center at
http://www.ibm.com/eserver/iseries/infocenter.

Output (OUTPUT)

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

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

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

*NONE

No output is created.

*PRINT

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

'stream-file-path-name'

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

'user-space-path-name'

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

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media
library device, on which the data is saved. The volumes must be placed in the device in the same order
as specified for this parameter.

A maximum of 75 volume identifiers can be specified. After all specified volumes are filled, the save
operation continues on whatever volumes are placed in the device.

Single values

*MOUNTED

The data is saved on the volumes placed in the device. For a media library device, the volume to
be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY)
command.

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

Other values (up to 75 repetitions)

character-value
> Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Label (LABEL)

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

*GEN The file label is created by the system.

- For objects in libraries, this is the equivalent of LABEL(*LIB) on the Save Object (SAVOBJ) and Save Library (SAVLIB) commands.
- For document library objects, this is the equivalent of LABEL(*GEN) on the Save Document Library Object (SAVDLO) command.
- For objects in other file systems, the label is **SAVyyyymmdd**.

character-value
> Specify the identifier (up to 17 characters) of the tape file used for the save operation.

Optical file (OPTFILE)

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

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

*   The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/*'
> The system generates an optical file name in the specified directory of the optical volume.

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

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215
> Specify the sequence number of the file to be used for the save operation.

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

*PERM
> The file is protected permanently.

date Specify the date when protection for the file ends.

End of media option (ENDOPT)

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

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

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

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

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

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

*YES The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

*NO The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing is complete.

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

*NONE
> No notification message is sent.

*WRKSTN
> The notification message is sent to the work station message queue.

path-name
> Specify the path name of the message queue to be used.

Type of output information (INFTYPE)

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

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

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

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

System (SYSTEM)

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

*LCL   Only local objects are processed.

*RMT   Only remote objects are processed.

*ALL   Both local and remote objects are processed.

Time period for last change (CHGPERIOD)

Specifies a date/time range. Objects that were last changed within that range will be saved.

Element 1: Start date

*ALL   No starting date is specified. All objects last changed prior to the ending date will be saved.

*LASTSAVE
> The objects that have changed since the last time they were saved with UPDHST(*YES) specified are saved. **Notes:**
>
> 1. If this value is specified, the value *ALL must be specified for all other elements of this parameter.
> 2. For local file systems, the system archive attribute is used. For remote file systems, the PC archive attribute is used.

date   Specify the date after which objects that have changed are to be saved. The date must be specified in job date format.

Element 2: Start time

*ALL   All times of day are included in the range.

time   Specify the time on the start date after which objects that have changed are to be saved.

> The time is specified in 24-hour format with or without a time separator as follows:
>
> - With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
> - Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

Note: Specifying an explicit time is valid only if the starting date is an explicit date.

Element 3: End date

*ALL   No ending date is specified. All objects changed since the starting date will be saved.

date   Specify the date before which objects that have changed are to be saved. The date must be specified in the job date format.

Element 4: End time

*ALL   All times of day are included in the range.

time   Specify a time on the end date before which objects that have changed are to be saved.

> The time is specified in 24-hour format with or without a time separator as follows:
>
> - With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
> - Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

Note: Specifying an explicit time is valid only if the ending date is an explicit date.

Object pre-check (PRECHK)

Specifies whether the save operation ends if any of the selected objects cannot be saved.

*NO   The save operation does not end. Objects that can be saved are saved.

*YES   The save operation ends. Nothing is saved unless all of the selected objects can be saved.

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

*CURRENT
> The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV   The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value
> Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Update history (UPDHST)

Specifies whether to update the save history on the objects saved with this save operation. The save history information is used when CHGPERIOD(*LASTSAVE) is specified on a later save operation.

Single values

*NO   This save operation will not be updated in the save history of the selected objects.

*YES   The save history is updated. For local file systems, the system save history is updated. For remote file systems, the PC save history is updated.

Other values (up to 2 repetitions)

*SYS   The system save history is updated.

*PC   The PC save history is updated.

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.

2. Clearing an optical volume does initialize it.

3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE
> None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*ALL   All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

*AFTER
> All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.
>
> **Note:** The *AFTER value is not valid for save files.

*REPLACE
> Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

*DEV If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

*NO No data compression is performed.

*YES If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

*LOW If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

> **Note:** This value is not valid for tape.

*MEDIUM
If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

> **Note:** This value is not valid for tape.

*HIGH
If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

> **Note:** This value is not valid for tape.

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

*DEV Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

> **Note:** If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

> If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

*NO Device data compaction is not performed.

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation.

*DFT The operation uses the ASPDEV value appropriate for the file system from which objects are being saved. For Integrated File System objects, *ALLAVL is used. For objects from the QSYS file system, the corresponding save command ASPDEV default is used.

*ALLAVL
The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and all available independent ASPs.

* The operation includes the system ASP, all basic user ASPs, and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS

The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP

If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Scan objects (SCAN)

Specifies whether objects will be scanned while being saved when exit programs are registered with any of the integrated file system scan-related exit points and whether objects that previously failed a scan should be saved.

The integrated file system scan-related exit points are:
- QIBM_QP0L_SCAN_OPEN - Integrated File System Scan on Open Exit Program
- QIBM_QP0L_SCAN_CLOSE - Integrated File System Scan on Close Exit Program

For details on these exit points, see the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Element 1: Scan during save

*NO Objects will not be scanned by the scan-related exit programs.

*YES Objects will be scanned according to the rules described in the scan-related exit programs.

Element 2: Save failed objects

*NOSAVFAILED

Objects that have either previously failed a scan or that fail a scan by a QIBM_QP0L_SCAN_OPEN exit program during this save will not be saved.

*SAVFAILED

Objects that have either previously failed a scan or that fail a scan during this save will be saved.

Examples

Example 1: Saving All Data Not in Libraries and Not Document Library Objects
```
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ(('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT))
```

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

Example 2: Saving Changes Since the Last Time the Objects Were Saved
```
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ(('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT))
      CHGPERIOD(*LASTSAVE)
```

This command saves all objects that are not in libraries, that are not document library objects, and that have changed since the last time they were saved with UPDHST(*YES) specified.

Example 3: Saving Data That Has Not Changed Since 1999

```
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
OBJ(('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT))
CHGPERIOD((*ALL *ALL '12/31/99'))
```

This command saves all objects that are not in libraries, that are not document library objects, and that have not changed since December 31, 1999.

Example 4: Saving All Objects in the Current Directory

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

This command uses the default value '*' on the OBJ parameter to save all objects in the current directory and its subdirectories. This example is not valid if the current directory is the root directory or if the current directory is in the QDLS file system.

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

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

Example 5: Omitting Objects During a Save Operation

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

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

```
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ(('/A') ('/A/B/C' *OMIT))
```

This command saves all objects in directory /A and its subdirectories, except those in directory /A/B/C.

Example 6: Saving a Library

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

This command saves library A on the tape device named TAP01.

Example 7: Saving Two Libraries

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

These two commands save two libraries. The first command saves library A and the second command saves library B.

Example 8: Saving All Files in a Library

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

This command saves all files in the library MYLIB on the tape device named TAP01.

Example 9: Saving Two Objects in a Library

```
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ('/QSYS.LIB/MYLIB.LIB/MYPGM.PGM')
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ('/QSYS.LIB/MYLIB.LIB/MYFILE.FILE')
```

These two commands save two objects in the same library. This first command saves the program MYPGM from library MYLIB. The second command saves the file MYFILE from library MYLIB.

Example 10: Saving a Stream File, a Database File, and a Document
```
SAV   DEV('/QSYS.LIB/TAP01.DEVD')  OBJ('/MyDir/MyFile')
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ('/QSYS.LIB/MYLIB.LIB/MYFILE.FILE')
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ('/QDLS/MYFLR/MYDOC')  SUBTREE(*OBJ)  UPDHST(*YES)
```

Three commands are used in this example to save three objects. The first command saves the stream file MyFile in the directory MyDir on the tape device named TAP01. The second command saves the database file MYFILE in the library named MYLIB on the tape device named TAP01. The third command saves the document MYDOC in a folder named MYFLR on the tape device named TAP01.

Example 11: Saving to a Save File
```
SAV   DEV('/QSYS.LIB/MYLIB.LIB/MYSAVF.FILE')  OBJ(MYDIR)
```

This command saves the directory MYDIR to a save file named MYSAVF.

Example 12: Using Symbolic Links for the Save Operation
```
SAV   DEV('DevLink')
      OBJ(('DirLink') ('FileLink') ('DirLink/*'))
      SAVACT(*YES)  SAVACTMSGQ('MsgqLink')
```

This command assumes that the current directory contains the following symbolic links:

- DevLink = /QSYS.LIB/TAP01.DEVD
- DirLink = /SomeDirectory
- FileLink = /SomeDirectory/SomeFile
- MsgqLink = /QSYS.LIB/LIB1.LIB/MSGQ1.MSGQ

This command saves the names associated with DirLink and FileLink, and the objects in SomeDirectory, to device TAP01. A message is sent when the save-while-active checkpoint is complete.

Symbolic links can be used to specify a device, a save-while-active message queue, and an output file. When symbolic links are specified to be saved, only the names of the associated objects are saved, not the content of the associated objects. A symbolic link to a directory can be used to save objects in the directory. Additional information about symbolic links is in the **Integrated file system** topic in the **File systems and management** category of the Information Center.

Error messages

*ESCAPE Messages

CPFA0DB
> Object not a QSYS.LIB object. Object is &1.

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

CPF3708
> Save file &1 in &2 too small.

CPF3727
> Duplicate device &1 specified on device name list.

CPF3735

Storage limit exceeded for user profile &1.

CPF3738

Device &1 used for save or restore is damaged.

CPF3768

Device &1 not valid for command.

CPF377D

Save ended because of read error on internal system resource.

CPF377E

Not enough storage for save-while-active request.

CPF378A

Message queue not available.

CPF378C

SAVACTMSGQ(*WRKSTN) not valid for batch job.

CPF3782

File &1 in &2 not a save file.

CPF3794

Save or restore operation ended unsuccessfully.

CPF380D

Save or restore of entire system completed unsuccessfully.

CPF3808

Save file &2 in &3 not complete.

CPF381D

Values on CHGPERIOD parameter not valid.

CPF3812

Save file &1 in &2 in use.

CPF382B

Parameters not valid with multiple file systems.

CPF382C

OBJ parameter value not valid for QSYS file system.

CPF382E

Specified parameter not valid for QDLS file system.

CPF382F

OBJ parameter value not valid for QDLS file system.

CPF3823

No objects saved or restored.

CPF3826

*INCLUDE object required on OBJ parameter.

CPF3828

Error occurred while attempting to use &1.

CPF3829

Specified parameter not valid for QSYS file system.

CPF383A

Save or restore ended unsuccessfully.

CPF383B

> End of file &1.

CPF383D

> Cannot use &1.

CPF3833

> Specified value on DEV parameter not valid.

CPF3834

> Too many values specified on the DEV parameter.

CPF3835

> Tape devices do not support same densities.

CPF3837

> &1 objects saved. &2 not saved.

CPF3838

> &1 objects saved. &2 objects not saved.

CPF384A

> Volume identifier &1 not valid.

CPF384B

> Optical file specified not valid.

CPF384C

> Error occurred during CCSID conversion.

CPF384E

> USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF386B

> Not able to save &1.

CPF386C

> Not able to save &1.

CPF3894

> Cancel reply received for message &1.

CPF38A5

> Error on the PATTERN parameter.

CPF5729

> Not able to allocate object &1.

CPF9802

> Not authorized to object &2 in &3.

CPF9825

> Not authorized to device &1.

Top

Save APAR Data (SAVAPARDTA)

The Save APAR Data (SAVAPARDTA) command allows the user to save information required for an Authorized Problem Analysis Report (APAR).

Restrictions:

- The following user profiles have private authorities to use the command:
  - QPGMR
  - QSYSOPR
  - QSRV
  - QSRVBAS

Parameters

Keyword	Description	Choices	Notes
PRBID	Problem identifier	*Character value*, *NEW	Required, Positional 1

Problem identifier (PRBID)

Specifies the identifier (ID) of the problem for which APAR data is to be saved.

*NEW An open problem log record is created to track this APAR.

character-value
    Specify the ID of the problem for which APAR data is to be saved.

Examples

```
SAVAPARDTA   PRBID(*NEW)
```

This command creates an open problem log for which APAR data is saved. The user selects the data to be saved by indicating the choices on a list display. This data is saved in an APAR library.

Error messages

*ESCAPE Messages

CPF2182

        Not authorized to library &1.

CPF39FA

        Problem &1 &2 &3 not found

CPF39FF

        SAVAPARDTA command can not be run

CPF39F2

        Cannot allocate library &1

CPF39F4

        No more APAR data can be saved for this problem

CPF39F5

        Query of problem &1 failed

CPF39F6

        Problem record could not be created

CPF39F7

        Could not create user space in APAR library

CPF39F8

        Problem &1 could not be associated with an APAR library

CPF39F9

        Problem &1 &2 &3 in use

Top

Save Configuration (SAVCFG)

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

The Save Configuration (SAVCFG) command saves all configuration and system resource management (SRM) objects without requiring a system in a restricted state. The information saved includes the following:

- Line descriptions
- Controller descriptions
- Device descriptions
- Mode descriptions
- Class-of-service descriptions
- Network interface descriptions
- Network server descriptions
- NetBIOS descriptions
- Connection lists
- Configuration lists
- Hardware resource data
- Token-ring adaptor data

Information saved can be restored with the Restore Configuration (RSTCFG) command.

Restrictions:

- You must have save system (*SAVSYS) special authority to run this command.
- System resource management (SRM) objects are not saved if a Work with Hardware Products (WRKHDWPRD) job is running at the same time.

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): *Name*	Required, Positional 1
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional
SEQNBR	Sequence number	1-16777215, ***END**	Optional
EXPDATE	File expiration date	*Date*, ***PERM**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
USEOPTBLK	Use optimum block	***YES**, *NO	Optional
SAVF	Save file	*Qualified object name*	Optional
	Qualifier 1: Save file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OPTFILE	Optical file	*Path name*, *****	Optional
CLEAR	Clear	***NONE**, *ALL, *AFTER, *REPLACE	Optional

Keyword	Description	Choices	Notes
PRECHK	Object pre-check	*NO, *YES	Optional
DTACPR	Data compression	*DEV, *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	*DEV, *NO	Optional
OUTPUT	Output	*NONE, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	Qualified object name	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
OUTMBR	Output member options	Element list	Optional
	Element 1: Member to receive output	Name, *FIRST	
	Element 2: Replace or add records	*REPLACE, *ADD	

Device (DEV)

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

This is a required parameter.

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

optical-device-name
>Specify the name of the optical device used for the save operation.

tape-media-library-device-name
>Specify the name of the tape media library device used for the save operation.

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

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED
>The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

Other values (up to 75 repetitions)

character-value
> Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215
> Specify the sequence number of the file to be used for the save operation.

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.

2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

*PERM
> The file is protected permanently.

date Specify the date when protection for the file ends.

End of media option (ENDOPT)

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

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

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

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

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

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

*YES The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

*NO The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

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

Qualifier 1: Save file

name   Specify the name of save file to be used.

Qualifier 2: Library

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

*CURLIB
    The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

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

Optical file (OPTFILE)

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

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

*   The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/*'
        The system generates an optical file name in the specified directory of the optical volume.

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE
　　　　None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*ALL　　All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

*AFTER
　　　　All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

　　　　Note: The *AFTER value is not valid for save files.

*REPLACE
　　　　Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Object pre-check (PRECHK)

Specifies whether the save configuration operation ends if any of the objects satisfy the following conditions:

- The objects were previously found to be damaged.
- The objects are locked by another job.
- The user does not have authority to save the objects.

*NO The save operation continues, saving only configuration and system resource management (SRM) objects that can be saved.

*YES The save operation ends before any data is written to the media if any configuration objects or system resource manager objects cannot be saved.

Top

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

*DEV If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

*NO No data compression is performed.

*YES If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

*LOW If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

  Note: This value is not valid for tape.

*MEDIUM

  If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

  Note: This value is not valid for tape.

*HIGH

  If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

  Note: This value is not valid for tape.

Top

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

*DEV Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

358 IBM Systems - iSeries: i5/OS Commands Starting with RSTSYSINF (Restore System Information)

Note: If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

*NO   Device data compaction is not performed.

Output (OUTPUT)

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

*NONE
    No output listing is created.

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

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

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

File to receive output (OUTFILE)

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

Qualifier 1: File to receive output

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

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Output member options (OUTMBR)

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

Element 1: Member to receive output

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

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

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

Element 2: Replace or add records

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

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

Examples

Example 1: Saving Objects

```
SAVCFG   DEV(TAP01)  CLEAR(*ALL)
```

This command saves system resource management objects (hardware resource data and token-ring adaptor data) and all configuration objects (including all line, controller, device, mode, class-of-service, and network descriptions, configuration lists, and connection lists). They are saved on the TAP01 tape drive. CLEAR(*ALL) automatically clears all uncleared tapes when they are encountered.

Example 2: Saving Objects to a Specific Tape

```
SAVCFG   DEV(TAP01)  VOL(ABC)
```

This command saves the SRM and configuration objects on the TAP01 tape drive, starting on the tape volume labeled ABC. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be put on the TAP01 tape drive is issued.

Error messages

*ESCAPE Messages

CPF2206
> User needs authority to do requested function on object.

CPF222E
> &1 special authority is required.

CPF3709
> Tape devices do not support same densities.

CPF3727

        Duplicate device &1 specified on device name list.

CPF3728

        Device &1 specified with other devices.

CPF3731

        Cannot use &2 &1 in library &3.

CPF3733

        &2 &1 in &3 previously damaged.

CPF3737

        Save and restore data area &1 not found.

CPF376D

        Not all configuration objects saved to save file &3.

CPF376E

        Not all configuration objects saved.

CPF3767

        Device &1 not found.

CPF3768

        Device &1 not valid for command.

CPF3782

        File &1 in &2 not a save file.

CPF3793

        Machine or ASP storage limit reached.

CPF3794

        Save or restore operation ended unsuccessfully.

CPF3812

        Save file &1 in &2 in use.

CPF384E

        USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF388B

        Optical file path name not valid.

CPF3894

        Cancel reply received for message &1.

CPF5729

        Not able to allocate object &1.

CPF9809

        Library &1 cannot be accessed.

CPF9812

        File &1 in library &2 not found.

CPF9845

        Error occurred while opening file &1.

CPF9846

        Error while processing file &1 in library &2.

CPF9847

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

CPF9850

>Override of printer file &1 not allowed.

CPF9851

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

CPF9860

>Error occurred during output file processing.

Save Changed Objects (SAVCHGOBJ)

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

The Save Changed Object (SAVCHGOBJ) command saves a copy of each changed object or group of objects located in the same library. When *ALL is specified for the **Objects (OBJ)** parameter, objects can be saved from all user libraries or from a list of libraries. When saving to a save file, only one library can be specified. For database files, only the changed members are saved.

Objects changed since the specified date and time are saved with the following exceptions:
- If OBJJRN(*NO) is specified, objects currently being journaled are not saved, unless journaling was started after the specified date and time. This ensures that changes made to an object before journaling starts are not lost (because they were not journaled in a journal receiver).
- Freed objects (programs, files, journal receivers, and so forth) are not saved.
- User-defined messages, job and output queue definitions, and logical file definitions are saved, but the contents of those objects are not saved. Logical file access paths are saved if ACCPTH(*YES) is specified. The contents of a data queue can be saved by specifying *DTAQ for the **Queue data (QDTA)** parameter.

Specified objects that were changed and the libraries where they reside remain locked during the save operation.

Saved objects can be restored with the Restore Object (RSTOBJ) command.

To determine the date and time that an object was changed, run the Display Object Description (DSPOBJD) command with DETAIL(*FULL) specified. For database file members that were changed, run the Display File Description (DSPFD) command.

The types of objects that can be saved by this command are listed in the OBJTYPE parameter description in "Commonly used parameters: Expanded descriptions" in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter. The system saves the changed objects by writing a copy of each one on tapes, optical volumes, or a save file. The description of each object is changed with the date, time, and place when it was last saved and is controlled by the UPDHST parameter.

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

Restrictions:
- You must either have save system (*SAVSYS) special authority or the following object authorities:
  - object existence (*OBJEXIST) authority for each object to be saved
  - execute (*EXECUTE) authority to each library objects are saved from
- If you do not have *SAVSYS special authority, only those changed objects that you have authority for are saved.
- When saving to a tape or optical device, you must have use (*USE) authority to the device description and device file. When saving to a save file, you must have object operational (*OBJOPR) and add (*ADD) authorities to the save file, and *EXECUTE authority to the library where the save file is located.

- When using a media definition, you must have *USE authority to the media definition and *EXECUTE authority to the media definition library.
- If tape is used, a standard labeled volume designation must be used.
- No changed object that is being saved can be changed by another job that is running when the save operation occurs unless save-while-active is used.
- When the contents of a save file are being saved to the same save file, by specifying SAVFDTA(*YES), only the description of the save file is saved.
- When the contents of a save file are saved with SAVFDTA(*YES), the save file must be restored before objects contained in it can be restored.
- When using the OUTFILE parameter to save to an existing database file, you must have *EXECUTE authority to the library where the file is located.
- When using the **Command user space (CMDUSRSPC)** parameter, you must have *USE authority to the user space and *EXECUTE authority to the user space library.

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 1
LIB	Library	Single values: *ALLUSR, *USRSPC Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 2
DEV	Device	Single values: *SAVF, *MEDDFN Other values (up to 4 repetitions): *Name*	Required, Positional 3
OBJTYPE	Object types	Single values: ***ALL** Other values (up to 300 repetitions): *Character value*	Optional
OBJJRN	Journaled objects	***NO**, *YES	Optional
REFDATE	Reference date	*Date*, ***SAVLIB**	Optional
REFTIME	Reference time	*Time*, ***NONE**	Optional
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional
SEQNBR	Sequence number	1-16777215, ***END**	Optional
LABEL	Label	*Character value*, ***LIB**	Optional
EXPDATE	File expiration date	*Date*, ***PERM**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
STRLIB	Starting library	*Name*, ***FIRST**	Optional
SAVF	Save file	*Qualified object name*	Optional
	Qualifier 1: Save file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MEDDFN	Media definition	*Qualified object name*	Optional
	Qualifier 1: Media definition	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OPTFILE	Optical file	*Path name*, *****	Optional
USEOPTBLK	Use optimum block	***YES**, *NO	Optional
TGTRLS	Target release	***CURRENT**, *PRV, V5R2M0, V5R3M0, V5R4M0	Optional
UPDHST	Update history	***YES**, *NO	Optional
CLEAR	Clear	***NONE**, *ALL, *AFTER, *REPLACE	Optional
PRECHK	Object pre-check	***NO**, *YES	Optional

Keyword	Description	Choices	Notes
SAVACT	Save active	***NO**, *LIB, *SYNCLIB, *SYSDFN	Optional
SAVACTWAIT	Save active wait time	*Element list*	Optional
	Element 1: Object locks	0-99999, **120**, *NOMAX	
	Element 2: Pending record changes	0-99999, ***LOCKWAIT**, *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, ***LOCKWAIT**, *NOMAX	
SAVACTMSGQ	Save active message queue	*Qualified object name*	Optional
	Qualifier 1: Save active message queue	*Name*, ***NONE**, *WRKSTN	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
ACCPTH	Save access paths	***SYSVAL**, *NO, *YES	Optional
SAVFDTA	Save file data	***YES**, *NO	Optional
QDTA	Queue data	***NONE**, *DTAQ	Optional
DTACPR	Data compression	***DEV**, *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	***DEV**, *NO	Optional
OMITLIB	Libraries to omit	Single values: ***NONE**, *USRSPC Other values (up to 300 repetitions): *Generic name, name*	Optional
OMITOBJ	Objects to omit	Single values: *USRSPC Other values (up to 300 repetitions): *Element list*	Optional
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Generic name, name*, ***NONE**, *ALL	
	Qualifier 2: Library	*Generic name, name*, ***ALL**	
	Element 2: Object type	*Character value*, ***ALL**	
ASPDEV	ASP device	*Name*, *****, *SYSBAS, *CURASPGRP	Optional
OUTPUT	Output	***NONE**, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	*Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	
INFTYPE	Type of output information	***OBJ**, *LIB, *MBR, *ERR	Optional
CMDUSRSPC	Command user space	*Qualified object name*	Optional
	Qualifier 1: Command user space	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Objects (OBJ)

Specifies the names of one or more objects, or the generic names of each group of objects, to check for changes and then to save those objects that have changed. All the objects must be in the library specified for the **Library (LIB)** parameter. If *ALL is specified or defaulted for the **Object types (OBJTYPE)** parameter, all the object types listed in the description of that parameter are saved, provided they are in the specified library and have the specified names.

This is a required parameter.

Single values

*ALL   All changed objects in the specified libraries are saved, depending on the values specified for the OBJTYPE parameter.

Other values (up to 300 repetitions)

generic-name
Specify one or more generic names of groups of changed objects to save in the specified library. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete object name.

name   Specify the names of specific objects to save. Both generic names and specific names can be specified in the same command.

Top

Library (LIB)

Specifies the library that contains the changed objects to be saved.

Note: Libraries are saved alphabetically by ASP device name for each value specified. Libraries on independent ASPs are saved before libraries on the system and basic user ASPs. For value *ALLUSR, libraries QSYS2, QGPL, QUSRSYS, and QSYS2xxxxx (where xxxxx is an independent ASP number) are saved first, if they are located on the ASPs specified by the ASPDEV parameter.

This is a required parameter.

Single values

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

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

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

```
QDSNX       QRCLxxxxx   QUSRIJS     QUSRVxRxMx
QGPL        QSRVAGT     QUSRINFSKR
QGPL38      QSYS2       QUSRNOTES
QMGTC       QSYS2xxxxx  QUSROND
QMGTC2      QS36F       QUSRPOSGS
QMPGDATA    QUSER38     QUSRPOSSA
```

```
QMQMDATA    QUSRADSM    QUSRPYMSVR
QMQMPROC    QUSRBRM     QUSRRDARS
QPFRDATA    QUSRDIRCL   QUSRSYS
QRCL        QUSRDIRDB   QUSRVI
```

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

*USRSPC
> The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are saved. Only one library can be specified in the user space when saving to a save file. Special value *SPLF cannot be specified in the user space.

Other values (up to 300 repetitions)

generic-name
> Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. A generic library name cannot be specified when saving to a save file.

name   If *ALL is specified for the **Objects (OBJ)** parameter, up to 300 library names can be specified. Only one library can be specified when saving to a save file.

Top

Device (DEV)

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

This is a required parameter.

Single values

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

*MEDDFN
> The save operation is done using the devices and media identified in the media definition specified for the **Media definition (MEDDFN)** parameter.

Other values

optical-device-name
> Specify the name of the optical device used for the save operation.

tape-media-library-device-name
> Specify the name of the tape media library device used for the save operation.

tape-device-name
> Specify the names of one or more tape devices used for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Save Changed Objects (SAVCHGOBJ)   **367**

Object types (OBJTYPE)

Specifies the types of system objects whose changes are saved. The object types saved are also the ones saved and restored by the Save Library (SAVLIB), Restore Object (RSTOBJ), and Restore Library (RSTLIB) commands. Data dictionaries and the associated files are saved only by using the SAVLIB command.

Single values

*ALL   Changes to all object types that are specified by name, and which are in the specified library, are saved.

Other values (up to 300 repetitions)

object-type
Specify the value for each of the types of objects that are saved, such as command (*CMD), file (*FILE), or program (*PGM).

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

Journaled objects (OBJJRN)

Specifies whether to save changed objects that are currently being journaled and that have been journaled since the date and time specified for the **Reference date (REFDATE)** and **Reference time (REFTIME)** parameters.

*NO   Objects being journaled are not saved. If journaling was started after the specified date and time, the changed objects or changed database file members are saved. The date and time of the last journal start operation can be shown by using the Display Object Description (DSPOBJD) command.

*YES   Objects whose changes are entered in a journal are saved.

Reference date (REFDATE)

Specifies the reference date. Objects that have been changed since this date are saved.

*SAVLIB
The objects that have been changed since the date of the last running of the Save Library (SAVLIB) command are saved. If the specified library was never saved, a message is issued and the library is not saved, but the operation continues.

date   Specify the reference date; objects that have been changed since this date are saved. If you specify a date later than the date of the running of this command, a message is issued and the operation ends. The date must be specified in the job date format.

Reference time (REFTIME)

Specifies the reference time. Objects that have been changed since this time on the specified date are saved.

*NONE

No explicit time is specified. Any objects changed since the date specified for the **Reference date (REFDATE)** parameter are saved.

time   Specify the reference time; objects that have been changed since this time on the specified date are saved. If *SAVLIB is specified for the REFDATE parameter, no reference time can be specified. If you specify a time later than the time of the running of this command, a message is issued and the operation ends.

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

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

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED

The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

Other values (up to 75 repetitions)

character-value

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

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END   The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215

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

Label (LABEL)

Specifies the name that identifies the data file on the tape volume that is to be used for the save operation. If this parameter is used on the save command, the same label must be specified on the restore command.

Note: You cannot specify *SAVLIB on this parameter, since it is a special value for the **Label (LABEL)** parameter of the restore command and would prevent you from restoring what you saved.

*LIB   The file label is created by the system using the name of the library specified for the **Library (LIB)** parameter.

character-value
> Specify the data file identifier of the data file used for the save operation. A maximum of 17 characters can be used. This option is valid only for a single-library save operation.

Top

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

*PERM
> The file is protected permanently.

date   Specify the date when protection for the file ends.

Top

End of media option (ENDOPT)

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

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

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

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

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

Top

Starting library (STRLIB)

Specifies the library with which to begin the save operation.

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

The basic steps for restarting a save operation are:

1. Check the job log to determine the library where the previous save operation failed. Find the last library saved, which is indicated by a successful completion message.
2. Load the next tape and ensure the tape is initialized.
3. Add the following to your original save command:

   ```
   STRLIB(library-name)  OMITLIB(library-name)
   ```

   where the *library-name* for the STRLIB and OMITLIB parameters is the last library successfully saved. This starts the save operation on the library after the last successfully saved library.

   To restore the libraries, you will need to perform a separate restore operation for each save operation that was performed.

*FIRST
   The save operation begins with the first library value specified for the **Library (LIB)** parameter. If the first value is a generic name or special value, the save operation begins with the first library that matches this value.

name   Specify the name of the library with which to begin the save operation.

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

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

Qualifier 1: Save file

name   Specify the name of save file to be used.

Qualifier 2: Library

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

*CURLIB
   The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

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

Media definition (MEDDFN)

Specifies the media definition (*MEDDFN) object that identifies the devices and media used to contain the saved data. For information about creating and using a media definition, see the Backup and Recovery book, SC41-5304, and the Create Media Definition API in the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

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

Qualifier 1: Media definition

name   Specify the name of the media definition to be used.

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

<div align="right">Top</div>

Optical file (OPTFILE)

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

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

*   The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/*'
> The system generates an optical file name in the specified directory of the optical volume.

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

<div align="right">Top</div>

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

*YES   The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

*NO   The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

<div align="right">Top</div>

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to restore and use the object.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

To specify that an object be saved for distribution to a system at a different release level than the system on which the save operation is to occur, the procedure differs for program or non-program objects and by the release level on which a program object is created. If, for example, you are saving an object for distribution to a target system running on an earlier release, you have the following choices:

For program objects:
- If the program object was created at a release level more current than the targeted earlier release, you must:
  1. create the program object again specifying the targeted earlier release
  2. save the program object specifying the targeted earlier release
  3. restore the program object on the target system.
- If the program object was created at the same release level as the target system, you can:
  1. save the program object specifying the targeted earlier release
  2. restore the program object on the target system.

For non-program objects:

You can:
1. save the object specifying the targeted earlier release
2. restore the object on the target system.

*CURRENT
> The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value
> Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Notes:
1. If LIB(*ALLUSR) is specified, only the current release can be the target release.
2. Not all objects can be targeted to another release. To find out which objects are supported, see the chart in the Backup and Recovery book, SC41-5304.

Update history (UPDHST)

Specifies whether the save history information of each saved object is changed with the date, time, and location of this save operation. The save history information for an object is displayed using the Display Object Description (DSPOBJD) command. The save history information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) or FROMENTLRG(*LASTSAVE) are used on the Apply Journaled Changes (APYJRNCHG) command.

*YES   The last save date, time, and location is updated in each object saved.

*NO   The save history information contained in the description of each object saved is not updated.

> **Note:** UPDHST(*NO) should be used for a save operation that is not intended for recovery. For example, if the save data is sent, record by record, to another system and the save file immediately deleted, the save history information is probably not to be updated.

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE
> None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*ALL   All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

*AFTER
> All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

> **Note:** The *AFTER value is not valid for save files.

*REPLACE
> Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:

1. The objects do not exist
2. The library or the objects were previously found to be damaged
3. The library or the objects are locked by another job
4. The requester of the save operation does not have authority for the library or to save the objects.

*NO   The save operation for a library continues, saving only those objects that can be saved.

*YES   If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is performed as if SAVACT(*NO) was specified.

*NO   Objects that are in use are not saved. Objects cannot be updated while being saved.

*LIB   Objects in a library can be saved while they are in use by another job. All of the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

*SYNCLIB
> Objects in a library can be saved while they are in use by another job. All of the objects and all of the libraries in the save operation reach a checkpoint together and are saved in a consistent state in relationship to each other.
>
> **Note:** If you specify this value and you are saving many libraries, it can take a long time to reach a checkpoint for all of the objects and libraries in the save operation.

*SYSDFN
> Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an object remains in use for the specified time, the object is not saved.

<u>120</u>
The system waits up to 120 seconds for each individual object lock before continuing the save operation.

*NOMAX
No maximum wait time exists.

0-99999
Specify the number of seconds to wait for each individual object lock before continuing the save operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)** parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value *NOCMTBDY is specified.

<u>***LOCKWAIT**</u>
The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

*NOCMTBDY
The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify *NOCMTBDY.

Note: This value cannot be specified if the **Target release (TGTRLS)** parameter value is earlier than V5R3M0.

*NOMAX
No maximum wait time exists.

0-99999
Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.

- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

*LOCKWAIT
>The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

*NOMAX
>No maximum wait time exists.

0-99999
>Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.
>
>If 0 is specified, and only one name is specified for the **Objects (OBJ)** parameter, and *FILE is the only value specified for the **Object types (OBJTYPE)** parameter, the system will save the object without requiring the types of transactions that are listed above to reach a commit boundary.

Top

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE
>No notification message is sent.

*WRKSTN
>The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name   Specify the name of the message queue to be used.

Qualifier 2: Library

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

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

name   Specify the name of the library where the message queue is located.

Top

Save access paths (ACCPTH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Backup and Recovery book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

*SYSVAL

> The QSAVACCPTH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

*NO Only those objects specified on the command are saved. No logical file access paths are saved.

*YES The specified physical files and all eligible logical file access paths over them are saved.

> **Note:** Specifying this value does not save the logical files.

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

*YES The description and contents of a save file are saved.

*NO Only the description of a save file is saved.

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

*NONE

> Only the description of a queue is saved.

*DTAQ

> The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

*DEV If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

*NO No data compression is performed.

*YES If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

*LOW If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

      Note: This value is not valid for tape.

*MEDIUM

      If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

      Note: This value is not valid for tape.

*HIGH

      If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

      Note: This value is not valid for tape.

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

*DEV Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

      Note: If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

      If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

*NO Device data compaction is not performed.

Libraries to omit (OMITLIB)

Specifies the names of one of more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

*NONE
> No libraries are excluded from the save operation.

*USRSPC
> The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

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

name   Specify the name of the library to be excluded from the save operation.

Objects to omit (OMITOBJ)

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

Single values

*USRSPC
> The objects identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

Element 1: Object

> **Qualifier 1: Object**

> ***NONE**
> > No objects are excluded from the operation.

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

> *generic-name*
> > Specify the generic name of the objects to be excluded.

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

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

Qualifier 2: Library

*ALL   The specified objects are excluded from all libraries that are part of the operation.

generic-name
> Specify the generic name of the libraries that contain objects to be excluded.

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

name   Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

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

character-value
> Specify the object type of the objects to be excluded from the operation.

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

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

*   The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS
> The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP
> If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name   Specify the name of the ASP device to be included in the save operation.

Output (OUTPUT)

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

*NONE
> No output listing is created.

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

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

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

File to receive output (OUTFILE)

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

Qualifier 1: File to receive output

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

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Output member options (OUTMBR)

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

Element 1: Member to receive output

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

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

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

Element 2: Replace or add records

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

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

Type of information (INFTYPE)

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

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

*ERR   The list contains information about the command, an entry for each library, and an entry for each object that was not successfully saved.

*LIB   The list contains a library entry for each library requested to be saved

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

Command user space (CMDUSRSPC)

Specifies a user space containing the values for the parameters which have *USRSPC specified for this command. The user space allows up to 32767 list values for each parameter, while the command parameters only allow up to 300 list values. The user space must define the parameters in the format used by the Save Object List (QSRSAVO) API.

Qualifier 1: User space

name   Specify the name of the user space containing the values for the parameters which have *USRSPC specified for this command.

Qualifier 2: Library

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

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

name   Specify the name of the library where the user space is located.

Examples

Example 1: Saving Changed Files
```
SAVCHGOBJ   OBJ(ORD*)  LIB(DSTPRODLB)  DEV(TAP01)
            OBJTYPE(*FILE)  REFDATE(122290)
```

This command saves all files with names that start with the characters ORD in the library named DSTPRODLB that were changed since December 22, 1990.

Example 2: Saving Description and Data for Save Files
```
SAVCHGOBJ   OBJ(FILE*)  LIB(MYLIB)  DEV(TAP01)  OBJTYPE(*FILE)
            REFDATE(122290)  SAVFDTA(*YES)
```

This command saves all files with names that start with the characters FILE* in the library named MYLIB that were changed since December 22, 1990. It also saves the description and the data for all save files that match this selection criteria.

Error messages

<u>*ESCAPE Messages</u>

CPF3702

&1 objects saved from &3. &2 not saved. &9 not included.

CPF3703

&2 &1 in &3 not saved.

CPF3708

Save file &1 in &2 too small.

CPF3709

Tape devices do not support same densities.

CPF3727

Duplicate device &1 specified on device name list.

CPF3728

Device &1 specified with other devices.

CPF3730

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

CPF3731

Cannot use &2 &1 in library &3.

CPF3733

&2 &1 in &3 previously damaged.

CPF3735

Storage limit exceeded for user profile &1.

CPF3738

Device &1 used for save or restore is damaged.

CPF3745

No record of SAVLIB operation exists for &1.

CPF3746

System date and time earlier than reference date and time.

CPF3747

Object names cannot be specified with more than one library.

CPF3749

Objects from library &2 not saved.

CPF3767

Device &1 not found.

CPF3768

Device &1 not valid for command.

CPF377D

Save ended because of read error on internal system resource.

CPF377E

Not enough storage for save-while-active request.

CPF377F

Save-while-active request prevented by pending record changes.

CPF3770

No objects saved or restored for library &1.

CPF3774

&1 objects saved from &3. &2 not saved. &8 not included.

CPF3778

Not all objects saved from all libraries.

CPF378A

Message queue not available.

CPF378C

SAVACTMSGQ(*WRKSTN) not valid for batch job.

CPF378E

Library &1 not saved.

CPF3781

Library &1 not found.

CPF3782

File &1 in &2 not a save file.

CPF3789

Only one library allowed with specified parameters.

CPF379E

Not enough storage available to save library &1.

CPF3793

Machine or ASP storage limit reached.

CPF3794

Save or restore operation ended unsuccessfully.

CPF3797

Objects from library &3 not saved. Save limit exceeded.

CPF37AB

*NOCMTBDY not allowed with target release.

CPF37AC

Library not allowed with *NOCMTBDY.

CPF37B4

User space &1 in &2 not valid.

CPF380B

Save cannot be completed at this time.

CPF3812

Save file &1 in &2 in use.

CPF3815

Save file &1 in &2 too small for save operation.

CPF3818

Starting library &1 not found.

CPF384E

USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3867

Contents of FILEMBR parameter not correct.

CPF3868

        FILEMBR specified but OBJTYPE must be *ALL or *FILE.

CPF3871

        No objects saved or restored; &3 objects not included.

CPF388B

        Optical file path name not valid.

CPF3892

        &2 &1 in &3 not saved.

CPF3894

        Cancel reply received for message &1.

CPF38A2

        ASP device &1 not correct.

CPF38A3

        File &1 in &2 not valid with ASPDEV.

CPF38A4

        ASP device &1 not correct.

CPF5729

        Not able to allocate object &1.

CPF9809

        Library &1 cannot be accessed.

CPF9812

        File &1 in library &2 not found.

CPF9814

        Device &1 not found.

CPF9820

        Not authorized to use library &1.

CPF9822

        Not authorized to file &1 in library &2.

CPF9825

        Not authorized to device &1.

CPF9833

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

CPFB8ED

        Device description &1 not correct for operation.

***STATUS Messages**

CPF3770

        No objects saved or restored for library &1.

CPF3774

        &1 objects saved from &3. &2 not saved. &8 not included.

CPF3871

        No objects saved or restored; &3 objects not included.

Top

Save Document Library Object (SAVDLO)

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

The Save Document Library Object (SAVDLO) command saves a copy of the specified documents, folders, or distribution objects (mail).

Notes:

- When a folder is saved, the folder object is saved along with the documents contained in that folder and the subfolders and documents in the subfolders and all successively nested folders and documents. Specific folders can be saved individually using DLO(*FLRLVL).

- Distribution objects (mail) cannot be saved or restored for individual users. Mail can be saved only for all users.

- SAVDLO does not require a dedicated system; however, individual objects in use when the save is issued cannot be saved. To ensure all document library objects are saved, run this command when no document or folder activity is occurring on the system.

Restrictions:

- You must have all object (*ALLOBJ) or save system (*SAVSYS) special authority to use the following parameter combinations on this command:
  - DLO(*ALL) FLR(*ANY)
  - DLO(*CHG)
  - DLO(*MAIL)
  - DLO(*SEARCH) OWNER(*ALL)
  - DLO(*SEARCH) OWNER(*user-profile-name*)

  where the *user profile name* specified is not the *user profile name* of the user issuing the SAVDLO command.

- If you do not have *ALLOBJ or *SAVSYS special authority, you must:
  - Have all (*ALL) authority for each document or folder to be saved
  - Be enrolled as a Document Interchange Architecture (DIA) user

- Determining document or folder ownership does not include checking group profiles if one is associated with the specified user profile.

- When using the OUTFILE parameter to save to an existing database file, you must have execute authority to the output file library.

Top

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Single values: *ALL, *SEARCH, *CHG, *SYSOBJNAM, *DOCL, *MAIL, *FLRLVL Other values (up to 300 repetitions): *Character value*	Required, Positional 1
FLR	Folder	Single values: ***ANY**, *NONE Other values (up to 300 repetitions): *Character value*	Optional

Keyword	Description	Choices	Notes
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): *Name*	Required, Positional 2
SRCHTYPE	Search type	***DOC**, *ALL	Optional
CHKFORMRK	Check for mark	***NO**, *YES	Optional
CHKEXP	Expiration date	*Date*, ***NO**, *CURRENT	Optional
CRTDATE	Creation date	*Element list*	Optional
	Element 1: Starting time and date	*Element list*	
	Element 1: Starting time	*Time*, ***AVAIL**	
	Element 2: Starting date	*Date*, ***BEGIN**, *CURRENT	
	Element 2: Ending time and date	*Element list*	
	Element 1: Ending time	*Time*, ***AVAIL**	
	Element 2: Ending date	*Date*, ***END**	
DOCCLS	Document class	*Character value*, ***ANY**	Optional
OWNER	Owner profile	*Name*, ***CURRENT**, *ALL	Optional
REFCHGDATE	Last changed date	*Date*, ***ANY**, *SAVDLOALL	Optional
REFCHGTIME	Last changed time	*Time*, ***ANY**	Optional
SYSOBJNAM	System object name	Values (up to 300 repetitions): *Name*	Optional
DOCL	Document list	*Character value*, ***NONE**	Optional
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional
SEQNBR	Sequence number	1-16777215, ***END**	Optional
EXPDATE	File expiration date	*Date*, ***PERM**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
LABEL	Label	*Character value*, ***GEN**	Optional
USEOPTBLK	Use optimum block	***YES**, *NO	Optional
OPTFILE	Optical file	*Path name*, *****	Optional
SAVF	Save file	*Qualified object name*	Optional
	Qualifier 1: Save file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OMITFLR	Folders to omit	Single values: ***NONE** Other values (up to 300 repetitions): *Character value*	Optional
OUTPUT	Output	***NONE**, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	*Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	
CLEAR	Clear	***NONE**, *AFTER, *ALL, *REPLACE	Optional
STG	Storage	***KEEP**, *DELETE, *FREE	Optional
DTACPR	Data compression	***DEV**, *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	***DEV**, *NO	Optional

Keyword	Description	Choices	Notes
CMDCHRID	Command character identifier	Single values: ***SYSVAL**, *DEVD Other values: *Element list*	Optional
	Element 1: Graphic character set	*Integer*	
	Element 2: Code page	*Integer*	
TGTRLS	Target release	***CURRENT**, *PRV, V5R2M0, V5R3M0, V5R4M0	Optional
SAVACT	Save active	***NO**, *YES	Optional
SAVACTWAIT	Save active wait time	0-99999, **120**, *NOMAX	Optional
ASP	ASP number	1-32, ***ANY**	Optional

Document library object (DLO)

Specifies the document library objects to save. To save a folder, DLO(*ALL) must be specified.

This is a required parameter.

Single values

*ALL   All document library objects further qualified by the FLR parameter are to be saved. Specifying DLO(*ALL) FLR(*ANY) saves all document library objects.

*SEARCH
> All document library objects that meet the specified search values are saved. Search values are specified by using the following parameters:
> - **Folder (FLR)** parameter.
> - **Check for mark (CHKFORMRK)** parameter.
> - **Expiration date (CHKEXP)** parameter.
> - **Creation date (CRTDATE)** parameter.
> - **Document class (DOCCLS)** parameter.
> - **Owner profile (OWNER)** parameter.
> - **Last changed date (REFCHGDATE)** parameter.
> - **Last changed time (REFCHGTIME)** parameter.
>
> **Note:** Folders are saved only if SRCHTYPE(*ALL) is specified.

*CHG   All documents created or changed and all folders created since the last complete save operation and all mail is saved.

*SYSOBJNAM
> The documents with the system object names specified for the **System object name (SYSOBJNAM)** parameter are saved.

*DOCL
> The list of documents referred to in a document list specified for the **Document list (DOCL)** parameter is saved.

*MAIL
> The distribution objects and documents referred to by a mail log are saved.

*FLRLVL
> The folders specified for the **Folder (FLR)** parameter and documents in the folders are saved. Subfolders are not saved.

Other values (up to 300 repetitions)

document-name
> Specify the user-assigned names of the documents that are saved. All documents specified must be in the same folder and that folder must be specified for the **Folder (FLR)** parameter.

Folder (FLR)

Specifies the name of the folder to save.

Single values

<u>*ANY</u> Document library objects can be saved from any folder. Consider the following when using the FLR parameter:
- FLR(*ANY) is not valid when one of the following is specified:
  - DLO(*DOCL)
  - DLO(*FLRLVL)
  - DLO(*document-name*)
- FLR(*ANY) is required when one of the following is specified:
  - DLO(*CHG)
  - DLO(*SYSOBJNAM)
  - DLO(*MAIL)
  - DLO(*SEARCH) SRCHTYPE(*ALL)
- When SAVDLO DLO(*ALL) FLR(*ANY) is specified, the following are saved:
  - All documents
  - All folders
  - All distribution objects (mail)

*NONE
> The documents saved are not in any folder. FLR(*NONE) is valid only when one of the following is specified:
- DLO(*ALL)
- DLO(*SEARCH) SRCHTYPE(*DOC)

Other values (up to 300 repetitions)

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

folder-name
> Specify the user-assigned name of the folder in which the documents to be saved are located. The folder name can be a maximum of 63 characters in length.
- Folder objects specified here are saved only when DLO(*ALL) or DLO(*FLRLVL) is specified.
- FLR(*folder-name*) is not valid when one of the following is specified:
  - DLO(*SYSOBJNAM)
  - DLO(*MAIL)
  - DLO(*SEARCH) SRCHTYPE(*ALL)

- Only one folder name can be specified when one of the following is specified:
  - DLO(*DOCL)
  - DLO(*SEARCH) SRCHTYPE(*DOC)
  - DLO(*document-name*)

Device (DEV)

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

This is a required parameter.

Single values

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

Other values

optical-device-name
> Specify the name of the optical device used for the save operation.

tape-media-library-device-name
> Specify the name of the tape media library device used for the save operation.

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

Search type (SRCHTYPE)

Specifies the type of objects for which to search. This parameter is valid only if *SEARCH is specified for the **Document library object (DLO)** parameter.

*DOC Only documents are to be searched and saved.

*ALL Documents and folders are to be searched and saved.

Check for mark (CHKFORMRK)

Specifies whether documents marked for storage are saved. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter and *DOC is specified for the **Search type (SRCHTYPE)** parameter.

*NO Documents that meet the other search values for this save operation are saved regardless of whether they are marked for storage.

*YES Only those documents that meet the other search values and are also marked for storage are saved. Documents may be marked:

- Keep
- Free
- Delete

Expiration date (CHKEXP)

Specifies that all documents with an expiration date before the specified date are to be saved. The expiration date is assigned by the user when a document is created to specify when the document is no longer needed. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter and *DOC is specified for the **Search type (SRCHTYPE)** parameter.

*NO   The expiration date is ignored.

*CURRENT
> All documents with an expiration date before today's date are saved.

date   Specify a document expiration date. All documents with an expiration date before this date are saved.

Creation date (CRTDATE)

Specifies that documents and folders that have a creation date during the time period specified are to be saved. The time period is specified by a starting time and date and an ending time and date. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter. The time period is specified as follows:

```
((start-time  start-date) (end-time  end-date))
```

Element 1: Starting time and date

Element 1: Starting time

Use one of the following to specify the starting time. Documents must have been created after this time to be selected. Documents created before this time are not selected.

*AVAIL
> Documents and folders filed at any time are eligible for saving.

time   Specify the starting time. When the starting time is used as a search value, the starting date must not be *BEGIN. The starting-time must be the same as the value specified for the **Last changed time (REFCHGTIME)** parameter when the REFCHGTIME parameter is specified. The time can be specified with or without a time separator:

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

Element 2: Starting date

Use one of the following to specify the starting date. Documents must have been created on or after this date to be saved. Documents created before this date are not saved.

*BEGIN
　　　　Documents and folders are saved regardless of the creation date associated with the object.

*CURRENT
　　　　Only documents and folders filed on today's date, after the starting time (if specified), are selected.

date　　Specify a starting date for the document creation date time period. The date must be entered in the job date format. The start date must be the same as the value specified for the **Last changed date (REFCHGDATE)** parameter when the REFCHGDATE parameter is specified.

Element 2: Ending time and date

Element 1: Ending time

Use one of the following to specify the ending time. Documents must have been created before this time to be saved. Any documents created after the specified time are not saved.

*AVAIL
　　　　Documents and folders filed at any time are selected for saving.

time　　Specify the ending time. When the ending time is to be used as a search value, the ending date must not be *END. See the description of *starting-time* for details about how time can be specified.

Element 2: Ending date

Use one of the following to specify the ending date. Documents must have been created on or before this date to be saved. Documents created after this date are not saved.

*END　Documents and folders filed on any date are selected. The ending time is not allowed when *END is specified.

date　　Specify the ending date for the document creation date time period. Documents created on or before this date are saved. The date must be specified in job date format.

Top

Document class (DOCCLS)

Specifies the class of documents to be saved. The class is assigned by the user when the document is created. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter and *DOC is specified for the **Search type (SRCHTYPE)** parameter.

Note: Although document classes are user-assigned, double-byte character set (DBCS) data cannot be specified on this parameter.

*ANY The document class is not used to select documents for saving.

character-value
> Specify the document class, ranging from 1 through 16 characters, used to select documents for saving.

Owner profile (OWNER)

Specifies the owner of the documents and folders to be saved. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter.

*CURRENT
> Documents and folders owned by the current requester are saved.

*ALL This parameter is not used to select documents and folders for saving. You must have all object (*ALLOBJ) or save system (*SAVSYS) special authority if *ALL is specified.

name Specify the name of the user profile that owns the documents and folders to be saved. All documents and folders owned by this user and that meet the other search values specified are saved. *ALLOBJ or *SAVSYS special authority is required if a user profile is selected other than the user profile of the user issuing this command.

Last changed date (REFCHGDATE)

Specifies the date after which the folders that are created and the documents that are changed or created are to be saved. The change date is updated when the document content or description is changed. This parameter is valid only when *SEARCH is specified for the **Document library object(DLO)** parameter.

*ANY No reference change date is specified. Documents are saved regardless of the date they were created or changed. Folders are saved regardless of the date they were created.

*SAVDLOALL
> Folders that have been created and documents created or changed since that last complete save operation are saved.

date Specify the date after which the created folders or the created or changed documents are saved.

Last changed time (REFCHGTIME)

Specifies the time, relative to the date specified for the REFCHGDATE parameter, after which the folders that are created and the documents that are changed or created are to be saved. The change time is updated when the document content or description is changed. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter.

*ANY No time is specified. The documents are saved regardless of the time they were created or changed. Folders are saved regardless of the time they were created.

time Specify the time after which the created folders or the created or changed documents are saved.

> The time can be specified with or without a time separator:
> - Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.

- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

System object name (SYSOBJNAM)

Specifies the system object names of the documents that are saved, when *SYSOBJNAM is specified for the **Document library object (DLO)** parameter.

Single values

*NONE
> A system object name is not specified.

Other values (up to 300 repetitions)

name   Specify the system object name of the document to be saved. A full ten characters must be specified.

Document list (DOCL)

Specifies a list of documents to be saved. The document list must be in a folder. The name of the folder must be specified using the **Folder (FLR)** parameter. You must have use (*USE) authority to the folders containing the documents in the document list.

Note: A document list must be the result of a local search, not a remote search.

*NONE
> No document list is saved.

character-value
> Specify the document list to be saved.

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED
> The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

Other values (up to 75 repetitions)

character-value
>   Specify the identifiers of one or more volumes in the order in which they are placed in a device
>   and used to save the data.

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END   The save operation begins after the last sequence number on the first tape. If the first tape is full,
>   an error message is issued and the operation ends.

1-16777215
>   Specify the sequence number of the file to be used for the save operation.

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is
protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.

2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

*PERM
>   The file is protected permanently.

date   Specify the date when protection for the file ends.

End of media option (ENDOPT)

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

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

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

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

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

Label (LABEL)

Specifies the name that identifies the data file on the tape used for the save. If the LABEL parameter is used the label must be specified on the restore command.

*GEN   The file label is created by the system.

character-value

Specify the data file identifier that is used as the label for the data file used for the save operation. A maximum of 17 characters can be specified.

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

*YES   The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

*NO   The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Optical file (OPTFILE)

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

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

*   The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/*'

The system generates an optical file name in the specified directory of the optical volume.

'optical-file-path-name'

Specify the path name of the optical file.

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

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

Qualifier 1: Save file

name   Specify the name of save file to be used.

Qualifier 2: Library

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

*CURLIB
>   The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

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

<div align="right">Top</div>

Folders to omit (OMITFLR)

Specifies the names of one of more folders, or the generic names of each group of folders, to be excluded from the save operation.

Single values

*NONE
>   No folders are excluded from the save operation.

Other values (up to 300 repetitions)

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

folder-name
>   Specify the name of the folder to be excluded from the save operation.

<div align="right">Top</div>

Output (OUTPUT)

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

*NONE
>   No output listing is created.

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

*OUTFILE

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

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

File to receive output (OUTFILE)

Specifies the name and library of the database file to which the information about the document library objects is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QAOJSAVO in QSYS with a format name of QJSDLO as a model.

Qualifier 1: File to receive output

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

Qualifier 2: Library

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

*CURLIB

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

name   Specify the name of the library to be searched.

Output member options (OUTMBR)

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

Element 1: Member to receive output

*FIRST

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

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

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

Element 2: Replace or add records

*REPLACE

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

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

Save Document Library Object (SAVDLO)   **399**

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE
None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*ALL All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

*AFTER
All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

*REPLACE
Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Storage (STG)

Specifies whether system storage occupied by the document being saved is kept, deleted, or freed after the save operation ends.

Note: STG(*DELETE) and STG(*FREE) are not valid when any of the following are specified:
- DLO(*ALL) FLR(*ANY)
- DLO(*SEARCH) CHKFORMRK(*YES)

- DLO(*CHG)
- DLO(*MAIL)
- SAVACT(*YES)

*KEEP
> The storage occupied by the document remains unchanged after the save operation.

*DELETE
> The document object and all search terms are deleted from the system after the save operation.

*FREE The document description and search terms remain on the system but the storage occupied by the document is deleted after the save operation. The document cannot be used until the document is restored to the system.

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

*DEV If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

*NO No data compression is performed.

*YES If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

*LOW If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

> **Note:** This value is not valid for tape.

*MEDIUM
> If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

> **Note:** This value is not valid for tape.

*HIGH
> If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

> **Note:** This value is not valid for tape.

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

*DEV Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

> **Note:** If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.
>
> If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

*NO Device data compaction is not performed.

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data specified for the **Document class (DOCCLS)** parameter. The character identifier is related to the display device used to enter the document class.

Single values

*SYSVAL

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

*DEVD

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

Element 1: Graphic character set

1-32767

> Specify the graphic character set to use.

Element 2: Code page

1-32767

> Specify the code page to use.

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

*CURRENT

> The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value

> Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

*NO Document library objects in use are not saved. Document library objects cannot be updated while being used.

*YES Document library objects can be changed during the save request.

> **Note:** Some applications update document library objects directly. The data is supplied to the application rather than saving the updates in a temporary file and then updating the DLOs. DLOs that are being updated directly (typically, those being updated by PC-based applications) will not be saved. See the Backup and Recovery information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information on using this parameter.

Save active wait time (SAVACTWAIT)

Specifies the amount of time for an object that is in use, before continuing the save operation. If an object remains in use for the specified time, the object is not saved.

120 The system waits up to 120 seconds for each individual object to become available before continuing the save operation.

*NOMAX

> No maximum wait time exists.

0-99999

> Specify the number of seconds to wait for each individual object before continuing the save operation.

ASP number (ASP)

Specifies the number of the auxiliary storage pool (ASP) of the document library object (DLO) to be saved.

*ANY The objects to be saved reside in any ASP. When DLO(*ALL) FLR(*ANY) are specified, all document library objects on the system are saved.

Note: When DLOs from multiple ASPs are saved, multiple tape media files are created. The beginning and ending sequence numbers of these media files will be required on the RSTDLO command to restore all ASPs.

1-32 Specify the number of an existing ASP that contains the document library objects to be saved. ASP 1 is the system ASP.

Note: Mail that has not been filed and documents that are not in a folder reside in the system ASP.

Examples

Example 1: Performing a Complete Save Operation

```
SAVDLO   DLO(*ALL)  FLR(*ANY)  DEV(TAP01)
```

This command saves all folders, documents, and mail to the tape device TAP01.

Example 2: Saving All Changes

```
SAVDLO   DLO(*CHG)  DEV(TAP01)
```

This command saves all documents created or changed since the last complete save operation, folders created since the last complete save operation, and all mail.

Example 3: Saving An Entire ASP

```
SAVDLO   DLO(*ALL)  DEV(TAP01)  FLR(*ANY)  ASP(2)
```

This command saves all folders and documents in user ASP 2.

Example 4: Saving Objects Changed After a Specific Date

```
SAVDLO   DLO(*SEARCH)  DEV(TAP01)  OWNER(*ALL)
         REFCHGDATE('01/01/2002')
```

This command saves all documents changed or created after January 1, 2002. This command is useful for saving changes between backups of the documents. This command is similar to the Save Changed Objects (SAVCHGOBJ) used for other object types.

Example 5: Saving Documents and Folders Changed After a Specific Date

```
SAVDLO   DLO(*SEARCH)  DEV(TAP01)  SRCHTYPE(*ALL)  OWNER(*ALL)
         REFCHGDATE('01/01/2002')
```

This command saves all folders created since 01/01/2002 and all documents created or changed since 01/01/2002.

Example 6: Saving Documents Created After a Specific Date

```
SAVDLO   DLO(*SEARCH)  DEV(TAP01)
         CRTDATE((*AVAIL '01/01/2002'))  OWNER(*ALL)
```

This command saves all documents created or changed since 01/01/2002.

Example 7: Saving Documents and Folders Created After a Specific Date

```
SAVDLO   DLO(*SEARCH)  DEV(TAP01)  SRCHTYPE(*ALL)
         CRTDATE((*AVAIL '01/01/2002'))  OWNER(*ALL)
```

This command saves all documents and folders created since 01/01/2002.

Example 8: Freeing System Storage During the Save Operation

```
SAVDLO   DLO(DOCX)   FLR(FOLDERA)   DEV(TAP01)   STG(*FREE)
```

This command saves the document named DOCX in folder FOLDERA to the tape device TAP01. As part of the save operation, the system storage that was occupied by the data portion of the document is freed.

Example 9: Saving Folders

```
SAVDLO   DLO(*ALL)   DEV(*SAVF)   FLR(BILL TOM MARY)   SAVF(SAVF1)
```

This command saves the folders BILL, TOM, and MARY, and all the documents in these folders.

Example 10: Saving Just the Folder Level of a Folder

```
SAVDLO   DLO(*FLRLVL)   FLR(DEC01)   DEV(*SAVF)   SAVF(MYLIB/SAVF1)
```

This command saves folder DEC01 in folder STATUS and all the documents in this folder to save file SAVF1 in library MYLIB. Subfolders in this folder are not saved.

Example 11: Saving Just Mail

```
SAVDLO   DLO(*MAIL)   DEV(*SAVF)
```

This command saves all distribution objects and all documents referred to by a mail log.

Error messages

*ESCAPE Messages

CPF3728
>	Device &1 specified with other devices.

CPF3733
>	&2 &1 in &3 previously damaged.

CPF3767
>	Device &1 not found.

CPF3768
>	Device &1 not valid for command.

CPF3793
>	Machine or ASP storage limit reached.

CPF3812
>	Save file &1 in &2 in use.

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

CPF384E
>	USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF8AC3
>	ASP &5 is not valid.

CPF8A47
>	Internal system objects in use.

CPF90B2
>	List of folder names not valid with DLO parameter.

CPF90C1

Document list &1 empty.

CPF90C2

Document list &1 not used.

CPF90D7

Documents not saved.

CPF90E1

Folder &1 cannot be used with TGTRLS(*PRV).

CPF90E5

Not authorized to document list &1.

CPF9006

User not enrolled in system distribution directory.

CPF902E

&1 document library objects saved. &10 not saved.

CPF903B

Too many objects for save file.

CPF903C

No document library objects saved.

CPF9030

Owner profile (&1) not found.

CPF9046

No documents found satisfying search specification in folder &1.

CPF9053

Not authorized to requested function.

CPF9056

Not authorized to requested function.

CPF906B

No document library objects saved.

CPF908A

Requester &1 not enrolled.

CPF9096

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

CPF941B

Save data area &1 in use.

CPF9410

&1 document library objects saved. &10 not saved.

CPF9411

Changes for ASP &1 not saved.

CPF9417

CRTDATE value not valid with REFCHGDATE or REFCHGTIME.

CPF9418

SAVF parameter not valid.

CPF9419

LABEL(*GEN) required.

CPF9810

Library &1 not found.

CPF9812

File &1 in library &2 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

CPF9825

Not authorized to device &1.

CPF9830

Cannot assign library &1.

CPF9831

Cannot assign device &1.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9850

Override of printer file &1 not allowed.

CPF9851

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

CPF9860

Error occurred during output file processing.

CPF9899

Error occurred during processing of command.

Top

Save Library (SAVLIB)

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

The Save Library (SAVLIB) command allows you to save a copy of one or more libraries.

When saving to a save file, only one library can be specified.

This command saves the entire library, including the library description, the object descriptions, and the contents of the objects in the library. For job queues, message queues, and logical files, only the object definitions are saved, not the contents. However, logical file access paths can be saved by specifying *YES for the **Save access paths (ACCPTH)** parameter. The contents of spooled files on output queues can be saved by specifying *ALL for the **Spooled file data (SPLFDTA)** parameter. The contents of a save file can be saved by specifying *YES for the **Save file data (SAVFDTA)** parameter or using the Save Save File Data (SAVSAVFDTA) command. The contents of a data queue can be saved by specifying *DTAQ for the **Queue data (QDTA)** parameter. The libraries and their objects are not affected in the system unless the command specifies that the storage is to be freed. However, unless *NO is specified for the **Update history (UPDHST)** parameter, the description of each library and each object is updated with the date, place, and time it was last saved. If a group of libraries is saved by specifying *NONSYS, *ALLUSR, or *IBM for the LIB parameter, the date, time, and place are updated in the history information for a data area in QSYS (data area QSAVLIBALL, QSAVALLUSR, or QSAVIBM).

The types of objects saved by this command are the same as those listed in the OBJTYPE parameter description in "Commonly used parameters: Expanded descriptions" in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter, with the addition of *DTADCT. Certain operating system objects that are not contained in user libraries (such as user profiles) are not saved by this command. They can be saved by the Save System (SAVSYS) or Save Security Data (SAVSECDTA) commands.

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

Restrictions:
- You must either have save system (*SAVSYS) special authority or the following object authorities:
  - read (*READ) authority for, or be the owner of, each library specified
  - object existence (*OBJEXIST) authority for each object in the library (for spooled file data, *OBJEXIST authority for the output queue)
- If you do not have *SAVSYS special authority, only those libraries and objects that you have authority for are saved.
- When saving to a tape or optical volume, you must have use (*USE) authority to the associated device description.
- When saving to a save file, you must have add (*ADD) and use (*USE) authorities to the save file.
- When using a media definition, you must have *USE authority to the media definition and execute (*EXECUTE) authority to the library where the media definition is located.
- When using the OUTFILE parameter to save to an existing database file, you must have *EXECUTE authority to the output file library.
- No library being saved, or the objects in it, can be updated by a job that is running at the time the save operation occurs unless save-while-active is used.
- When the contents of a save file are being saved to the same save file by specifying SAVFDTA(*YES), only the description of the save file is saved.

- When the contents of a save file are saved, by specifying *YES for the **Save file data (SAVFDTA)** parameter, the save file must be restored before objects contained in it can be restored.
- When using the **Command user space (CMDUSRSPC)** parameter, you must have *USE authority to the user space and *EXECUTE authority to the user space library.

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Single values: *NONSYS, *ALLUSR, *IBM, *USRSPC Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 1
DEV	Device	Single values: *SAVF, *MEDDFN Other values (up to 4 repetitions): *Name*	Required, Positional 2
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional, Positional 3
SEQNBR	Sequence number	1-16777215, ***END**	Optional
LABEL	Label	*Character value*, ***LIB**	Optional
EXPDATE	File expiration date	*Date*, ***PERM**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
STRLIB	Starting library	*Name*, ***FIRST**	Optional
SAVF	Save file	*Qualified object name*	Optional
	Qualifier 1: Save file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MEDDFN	Media definition	*Qualified object name*	Optional
	Qualifier 1: Media definition	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OPTFILE	Optical file	*Path name*, *****	Optional
USEOPTBLK	Use optimum block	***YES**, *NO	Optional
TGTRLS	Target release	***CURRENT**, *PRV, V5R2M0, V5R3M0, V5R4M0	Optional
UPDHST	Update history	***YES**, *NO	Optional
CLEAR	Clear	***NONE**, *ALL, *AFTER, *REPLACE	Optional
PRECHK	Object pre-check	***NO**, *YES	Optional
SAVACT	Save active	***NO**, *LIB, *SYNCLIB, *SYSDFN	Optional
SAVACTWAIT	Save active wait time	*Element list*	Optional
	Element 1: Object locks	0-99999, **120**, *NOMAX	
	Element 2: Pending record changes	0-99999, ***LOCKWAIT**, *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, ***LOCKWAIT**, *NOMAX	
SAVACTMSGQ	Save active message queue	*Qualified object name*	Optional
	Qualifier 1: Save active message queue	*Name*, ***NONE**, *WRKSTN	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
ACCPTH	Save access paths	***SYSVAL**, *NO, *YES	Optional
SAVFDTA	Save file data	***YES**, *NO	Optional
SPLFDTA	Spooled file data	***NONE**, *ALL	Optional
QDTA	Queue data	***NONE**, *DTAQ	Optional
STG	Storage	***KEEP**, *FREE	Optional

Keyword	Description	Choices	Notes
DTACPR	Data compression	***DEV**, *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	***DEV**, *NO	Optional
OMITLIB	Libraries to omit	Single values: ***NONE**, *USRSPC Other values (up to 300 repetitions): *Generic name, name*	Optional
OMITOBJ	Objects to omit	Single values: *USRSPC Other values (up to 300 repetitions): *Element list*	Optional
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Generic name, name*, ***NONE**, *ALL	
	Qualifier 2: Library	*Generic name, name*, ***ALL**	
	Element 2: Object type	*Character value*, ***ALL**	
ASPDEV	ASP device	*Name*, *****, *SYSBAS, *CURASPGRP	Optional
OUTPUT	Output	***NONE**, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	*Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	
INFTYPE	Type of output information	***OBJ**, *LIB, *MBR, *ERR	Optional
CMDUSRSPC	Command user space	*Qualified object name*	Optional
	Qualifier 1: Command user space	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Library (LIB)

Specifies which libraries are saved.

Notes:

1. The system libraries QDOC, QDOCxxxx, QRCYxxxxx, QRECOVERY, QRPLOBJ, QRPLxxxxx, QSPL, QSPLxxxx, QSRV, QSYS, QSYSxxxxx, and QTEMP cannot be saved with this command. xxxx is a basic user ASP number, and xxxxx is an independent ASP number.

2. If you specify *ALLUSR or *IBM on this parameter, this command should be run when the specified libraries are not being used. If objects in a library are in use while the library is being saved, the objects are not saved unless you use SAVACT. To ensure a complete save of all libraries, run this command with the system in a restricted state. For example, if SAVLIB LIB(*ALLUSR) is run when the subsystem QSNADS is active, the QAO* files are not saved in library QUSRSYS. To save the QAO* files, end the QSNADS subsystem before running SAVLIB LIB(*ALLUSR).

3. Doing a SAVLIB LIB(*IBM) and then doing a SAVLIB LIB(*ALLUSR) saves the same libraries as a SAVLIB LIB(*NONSYS), but requires two restore commands.

4. Libraries are saved alphabetically by ASP device name for each value specified. Libraries on independent ASPs are saved before libraries on the system and basic user ASPs. For values *NONSYS and *ALLUSR, libraries QSYS2, QGPL, QUSRSYS, and QSYS2xxxxx (where xxxxx is an independent ASP number) are saved first, if they are located on the ASPs specified by the ASPDEV parameter.

This is a required parameter.

Single values

*NONSYS

All libraries except the system libraries are saved. All subsystems must be ended by the End Subsystem (ENDSBS) or End System (ENDSYS) command before this option is specified.

*ALLUSR

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

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

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

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

*IBM Saves all IBM-supplied libraries except for the following:

```
QDOC        QRCYxxxxx   QTEMP       QUSRRDARS
QDOCxxxx    QRECOVERY   QUSER38     QUSRSYS
QDSNX       QRPLOBJ     QUSRADSM    QUSRVI
QGPL        QRPLxxxxx   QUSRBRM     QUSRVxRxMx
QGPL38      QSPL        QUSRDIRCL
QMGTC       QSPLxxxx    QUSRDIRDB
QMGTC2      QSRV        QUSRIJS
QMPGDATA    QSRVAGT     QUSRINFSKR
QMQMDATA    QSYS        QUSRNOTES
QMQMPROC    QSYSxxxxx   QUSROND
QPFRDATA    QSYS2       QUSRPOSGS
QRCL        QSYS2xxxxx  QUSRPOSSA
QRCLxxxxx   QS36F       QUSRPYMSVR
```

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

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

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

*USRSPC

The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are saved. Only one library can be specified in the user space when saving to a save file. Special value *SPLF cannot be specified in the user space.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries to be saved. A generic name is a character string of one

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

name   Specify the name of the library to be saved. Only one library name can be saved when saving to a save file. A system library name cannot be specified.

Device (DEV)

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

This is a required parameter.

Single values

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

*MEDDFN
        The save operation is done using the devices and media identified in the media definition specified for the **Media definition (MEDDFN)** parameter.

Other values

optical-device-name
        Specify the name of the optical device used for the save operation.

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

tape-device-name
        Specify the names of one or more tape devices used for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED
        The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

Other values (up to 75 repetitions)

character-value
> Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215
> Specify the sequence number of the file to be used for the save operation.
>
> If *NONSYS, *ALLUSR, or *IBM is specified for the LIB parameter, the save operation for the set of libraries begins at the sequence number specified. The first file saved in this set is the QFILE file. The QFILE file contains the list of libraries saved.

Label (LABEL)

Specifies the name that identifies the data file on the tape volume that is to be used for the save operation. If this parameter is used on the save command, the same label must be specified on the restore command.

Note: You cannot specify *SAVLIB on this parameter, since it is a special value for the **Label (LABEL)** parameter of the restore command and would prevent you from restoring what you saved.

*LIB The file label is created by the system using the name of the library specified for the **Library (LIB)** parameter.

character-value
> Specify the data file identifier of the data file used for the save operation. A maximum of 17 characters can be used. This option is valid only for a single-library save operation.

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

*PERM
> The file is protected permanently.

date Specify the date when protection for the file ends.

End of media option (ENDOPT)

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

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

*REWIND

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

*LEAVE

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

*UNLOAD

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

Starting library (STRLIB)

Specifies the library with which to begin the save operation.

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

The basic steps for restarting a save operation are:

1. Check the job log to determine the library where the previous save operation failed. Find the last library saved, which is indicated by a successful completion message.
2. Load the next tape and ensure the tape is initialized.
3. Add the following to your original save command:

   ```
   STRLIB(library-name)  OMITLIB(library-name)
   ```

   where the *library-name* for the STRLIB and OMITLIB parameters is the last library successfully saved. This starts the save operation on the library after the last successfully saved library.

   To restore the libraries, you will need to perform a separate restore operation for each save operation that was performed.

*FIRST

> The save operation begins with the first library value specified for the **Library (LIB)** parameter. If the first value is a generic name or special value, the save operation begins with the first library that matches this value.

name Specify the name of the library with which to begin the save operation.

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

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

Qualifier 1: Save file

name   Specify the name of save file to be used.

Qualifier 2: Library

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

*CURLIB
    The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

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

Media definition (MEDDFN)

Specifies the media definition (*MEDDFN) object that identifies the devices and media used to contain the saved data. For information about creating and using a media definition, see the Backup and Recovery book, SC41-5304, and the Create Media Definition API in the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

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

Qualifier 1: Media definition

name   Specify the name of the media definition to be used.

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Optical file (OPTFILE)

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

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

*   The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/'*
>   The system generates an optical file name in the specified directory of the optical volume.

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

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

*YES   The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:
   - Performance may improve.
   - The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
   - The value for the DTACPR parameter is ignored.

*NO   The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to restore and use the object.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

To specify that an object be saved for distribution to a system at a different release level than the system on which the save operation is to occur, the procedure differs for program or non-program objects and by the release level on which a program object is created. If, for example, you are saving an object for distribution to a target system running on an earlier release, you have the following choices:

For program objects:
- If the program object was created at a release level more current than the targeted earlier release, you must:
  1. create the program object again specifying the targeted earlier release
  2. save the program object specifying the targeted earlier release
  3. restore the program object on the target system.
- If the program object was created at the same release level as the target system, you can:
  1. save the program object specifying the targeted earlier release

2. restore the program object on the target system.

For non-program objects:

You can:

1. save the object specifying the targeted earlier release
2. restore the object on the target system.

<u>*CURRENT</u>
> The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value
> Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Notes:

1. If LIB(*NONSYS), LIB(*ALLUSR), or LIB(*IBM) is specified, only the current release can be the target release.
2. Not all objects can be targeted to another release. To find out which objects are supported, see the chart in the Backup and Recovery book, SC41-5304.

Update history (UPDHST)

Specifies whether the save history information of each saved object is changed with the date, time, and location of this save operation. The save history information for an object is displayed using the Display Object Description (DSPOBJD) command. The save history information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) or FROMENTLRG(*LASTSAVE) are used on the Apply Journaled Changes (APYJRNCHG) command.

<u>*YES</u> The last save date, time, and location is updated in each object saved.

*NO The save history information contained in the description of each object saved is not updated.

> **Note:** UPDHST(*NO) should be used for a save operation that is not intended for recovery. For example, if the save data is sent, record by record, to another system and the save file immediately deleted, the save history information is probably not to be updated.

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE
  None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*ALL All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

*AFTER
  All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

  Note: The *AFTER value is not valid for save files.

*REPLACE
  Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:
1. The objects do not exist
2. The library or the objects were previously found to be damaged
3. The library or the objects are locked by another job
4. The requester of the save operation does not have authority for the library or to save the objects.

*NO The save operation for a library continues, saving only those objects that can be saved.

*YES If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is performed as if SAVACT(*NO) was specified.

*NO   Objects that are in use are not saved. Objects cannot be updated while being saved.

*LIB   Objects in a library can be saved while they are in use by another job. All of the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

*SYNCLIB

Objects in a library can be saved while they are in use by another job. All of the objects and all of the libraries in the save operation reach a checkpoint together and are saved in a consistent state in relationship to each other.

Note: If you specify this value and you are saving many libraries, it can take a long time to reach a checkpoint for all of the objects and libraries in the save operation.

*SYSDFN

Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Top

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an object remains in use for the specified time, the object is not saved.

120   The system waits up to 120 seconds for each individual object lock before continuing the save operation.

*NOMAX

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for each individual object lock before continuing the save operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)** parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value *NOCMTBDY is specified.

*LOCKWAIT

The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

*NOCMTBDY

The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify *NOCMTBDY.

Note: This value cannot be specified if the **Target release (TGTRLS)** parameter value is earlier than V5R3M0.

*NOMAX
> No maximum wait time exists.

0-99999
> Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.
- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

*LOCKWAIT
> The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

*NOMAX
> No maximum wait time exists.

0-99999
> Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE
> No notification message is sent.

*WRKSTN
> The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name   Specify the name of the message queue to be used.

Qualifier 2: Library

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

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

name   Specify the name of the library where the message queue is located.

Save access paths (ACCPTH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Backup and Recovery book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

*SYSVAL
> The QSAVACCPTH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

*NO   Only those objects specified on the command are saved. No logical file access paths are saved.

*YES   The specified physical files and all eligible logical file access paths over them are saved.

> **Note:** Specifying this value does not save the logical files.

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

*YES   The description and contents of a save file are saved.

*NO   Only the description of a save file is saved.

Spooled file data (SPLFDTA)

Specifies whether to save spooled file data and attributes for output queues that are saved.

*NONE
 No spooled file data is saved.

*ALL   For each output queue that is saved, all available spooled file data on the output queue is saved.

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

*NONE
 Only the description of a queue is saved.

*DTAQ
 The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Storage (STG)

Specifies whether the system storage that is occupied by the data portion of the specified members (except for save files), modules, programs, service programs, Structured Query Language (SQL) packages, and journal receivers in the library being saved is freed as part of the save operation. Only the data portion of the objects is freed, not the descriptions of the objects.

*KEEP
 The storage occupied by the data portion of the objects being saved is not freed.

*FREE The storage occupied by the data portion of the specified objects being saved is freed as part of the save operation. The storage for all the objects in a library is freed only after all the objects in that library are saved successfully.

 Note: To prevent the possible abnormal end of a program, the program being saved must not be running in the system when *FREE is specified.

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

*DEV If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

*NO No data compression is performed.

*YES If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

*LOW If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

Note: This value is not valid for tape.

*MEDIUM
If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

Note: This value is not valid for tape.

*HIGH
If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

Note: This value is not valid for tape.

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

*DEV Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

Note: If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

*NO Device data compaction is not performed.

Libraries to omit (OMITLIB)

Specifies the names of one of more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

*NONE
> No libraries are excluded from the save operation.

*USRSPC
> The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

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

name Specify the name of the library to be excluded from the save operation.

Objects to omit (OMITOBJ)

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

Single values

*USRSPC
> The objects identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

Element 1: Object

Qualifier 1: Object

*NONE
> No objects are excluded from the operation.

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

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

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

Qualifier 2: Library

*ALL The specified objects are excluded from all libraries that are part of the operation.

generic-name
> Specify the generic name of the libraries that contain objects to be excluded.

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

name   Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

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

character-value

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

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

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

*   The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS

The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP

If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name   Specify the name of the ASP device to be included in the save operation.

Top

Output (OUTPUT)

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

*NONE

No output listing is created.

*PRINT

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

*OUTFILE

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

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

File to receive output (OUTFILE)

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

Qualifier 1: File to receive output

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

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Output member options (OUTMBR)

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

Element 1: Member to receive output

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

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

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

Element 2: Replace or add records

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

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

Type of information (INFTYPE)

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

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

*ERR   The list contains information about the command, an entry for each library, and an entry for each object that was not successfully saved.

*LIB   The list contains a library entry for each library requested to be saved

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

Command user space (CMDUSRSPC)

Specifies a user space containing the values for the parameters which have *USRSPC specified for this command. The user space allows up to 32767 list values for each parameter, while the command parameters only allow up to 300 list values. The user space must define the parameters in the format used by the Save Object List (QSRSAVO) API.

Qualifier 1: User space

name   Specify the name of the user space containing the values for the parameters which have *USRSPC specified for this command.

Qualifier 2: Library

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

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

name   Specify the name of the library where the user space is located.

Examples

Example 1: Saving a Library on a Tape Device
```
SAVLIB   LIB(JOE)  DEV(TAP01)
```

This command saves the library named JOE on the tape that is in the tape device named TAP01. The storage occupied by JOE in the system is not freed, because the STG parameter default (*KEEP) was assumed.

Example 2: Saving on Multiple Volumes
```
SAVLIB   LIB(QGPL)  DEV(OPT01)  VOL(ABC DEF GHI)
```

The general purpose library (QGPL) is saved on the optical volumes in the device named OPT01. The optical volumes used must have the volume names ABC, DEF, and GHI. If the save operation is not finished when volume ABC is full, a message is issued to the operator asking for volume DEF to be placed in the device.

Example 3: Freeing Storage when Saving Data
```
SAVLIB   LIB(CUSTDATA)  DEV(TAP01)
         VOL(CUSTNM CUSTAD)  STG(*FREE)
```

The library named CUSTDATA is saved on volumes CUSTNM and CUSTAD, which are put in the tape device TAP01. The storage occupied by the specified members, modules, programs, service programs, SQL packages, and journal receivers in the CUSTDATA library is freed after it is saved.

Example 4: Saving on Multiple Devices Serially

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

The library named USRLIB is saved on four tape volumes on three tape devices. The volume named USRA is put on the device named TAP01, the volume named USRB on the device named TAP02, the volume named USRC on the device named TAP03, and the volume named USRD on the device named TAP01. The volume named USRA is rewound, and must be unloaded by the operator when processing is complete so that the device named TAP01 can be used for the volume named USRD.

Example 5: Saving on Multiple Devices in Parallel

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

The library named USRLIB is saved on the devices specified in the media definition named MEDDFNA in library LIBA. For information about creating and using a media definition, see the **Backup, Recovery, and Availability** topic in the Information Center.

Example 6: Saving a Library with a Media File Label

```
SAVLIB   LIB(LIB1)  DEV(TAP01)  LABEL(MONDAYBACKUP)
```

This command uses the tape device named TAP01 to save the library named LIB1 on tape. The library is saved with a media file label of MONDAYBACKUP. This label must be specified when restoring the library or any of its objects.

Example 7: Specifying Where the Save Operation Starts

```
SAVLIB   LIB(*NONSYS)  DEV(TAP01 TAP02)
         STRLIB(MIKESLIB)  ASPDEV(*SYSBAS)
```

This command saves all libraries that are located on the system and basic user ASPS, beginning with the library named MIKESLIB on tape devices named TAP01 and TAP02. System libraries and libraries that are normally saved first (QSYS2, QGPL, and QUSRSYS) are not saved.

Example 8: Saving User Libraries in an Independent ASP Group

```
SAVLIB   LIB(*ALLUSR)  DEV(TAP01)  ASPDEV(*CURASPGRP)
```

This command saves all user libraries on the independent ASPs that are in the ASP group for the current thread. It does not save libraries on the system ASP (ASP number 1) or on basic user ASPs (ASP numbers 2-32).

Example 9: Saving Spooled Files

```
SAVLIB   LIB(MYLIB)  DEV(TAP01)  SPLFDTA(*ALL)
```

This command saves the library named MYLIB, including the spooled file data for any output queues in the library.

Error messages

*ESCAPE Messages

CPF3701

&1 objects saved from &3. &2 not saved.

CPF3703

&2 &1 in &3 not saved.

CPF3708

>Save file &1 in &2 too small.

CPF3709

>Tape devices do not support same densities.

CPF3727

>Duplicate device &1 specified on device name list.

CPF3728

>Device &1 specified with other devices.

CPF3730

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

CPF3731

>Cannot use &2 &1 in library &3.

CPF3733

>&2 &1 in &3 previously damaged.

CPF3735

>Storage limit exceeded for user profile &1.

CPF3738

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

CPF3749

>Objects from library &2 not saved.

CPF3751

>Some libraries not saved.

CPF3767

>Device &1 not found.

CPF3768

>Device &1 not valid for command.

CPF377D

>Save ended because of read error on internal system resource.

CPF377E

>Not enough storage for save-while-active request.

CPF377F

>Save-while-active request prevented by pending record changes.

CPF3770

>No objects saved or restored for library &1.

CPF3771

>&1 objects saved from &3. &2 not saved.

CPF3777

>&1 libraries saved, &6 partially saved, &2 not saved.

CPF378A

>Message queue not available.

CPF378C

>SAVACTMSGQ(*WRKSTN) not valid for batch job.

CPF378E

>Library &1 not saved.

CPF3781

   Library &1 not found.

CPF3782

   File &1 in &2 not a save file.

CPF3785

   Not all subsystems ended.

CPF3789

   Only one library allowed with specified parameters.

CPF379E

   Not enough storage available to save library &1.

CPF3793

   Machine or ASP storage limit reached.

CPF3794

   Save or restore operation ended unsuccessfully.

CPF3797

   Objects from library &3 not saved. Save limit exceeded.

CPF37AB

   *NOCMTBDY not allowed with target release.

CPF37AC

   Library not allowed with *NOCMTBDY.

CPF37B1

   SPLFDTA not allowed with target release.

CPF37B4

   User space &1 in &2 not valid.

CPF380B

   Save cannot be completed at this time.

CPF3812

   Save file &1 in &2 in use.

CPF3815

   Save file &1 in &2 too small for save operation.

CPF3818

   Starting library &1 not found.

CPF384E

   USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3871

   No objects saved or restored; &3 objects not included.

CPF388B

   Optical file path name not valid.

CPF3892

   &2 &1 in &3 not saved.

CPF3894

   Cancel reply received for message &1.

CPF38A2

   ASP device &1 not correct.

CPF38A3

       File &1 in &2 not valid with ASPDEV.

CPF38A4

       ASP device &1 not correct.

CPF5729

       Not able to allocate object &1.

CPF9809

       Library &1 cannot be accessed.

CPF9812

       File &1 in library &2 not found.

CPF9814

       Device &1 not found.

CPF9820

       Not authorized to use library &1.

CPF9822

       Not authorized to file &1 in library &2.

CPF9825

       Not authorized to device &1.

CPF9833

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

CPFB8ED

       Device description &1 not correct for operation.

OPT1332

       Optical volume not found in device &1.

<u>***STATUS Messages**</u>

CPF3770

       No objects saved or restored for library &1.

CPF3771

       &1 objects saved from &3. &2 not saved.

CPF3871

       No objects saved or restored; &3 objects not included.

Top

Save Licensed Program (SAVLICPGM)

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

The Save Licensed Program (SAVLICPGM) command saves a copy of all of the objects that make up a licensed program. It saves the licensed program in a form that can be restored by the Restore Licensed Program (RSTLICPGM) command.

Restrictions:

1. To use this command, you must have *ALLOBJ authority or have specific authority from the security officer.
2. A standard labeled tape volume must be on the tape device.
3. Some licensed programs can only be saved if the user is enrolled in the system distribution directory. See the publication for each licensed program for a description of this restriction.
4. This command does not save code or language objects for the base operating system.
5. At most one optical device, one save file, one virtual tape device or one tape media library device can be specified.

Parameters

Keyword	Description	Choices	Notes
LICPGM	Product	*Character value*	Required, Positional 1
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): *Name*	Required, Positional 2
OPTION	Optional part to be saved	***BASE**, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 01, 02, 03, 04, 05, 06, 07, 08, 09	Optional
RLS	Release	*Character value*, ***ONLY**	Optional
LNG	Language for licensed program	*Character value*, ***PRIMARY**, *ALL	Optional
OBJTYPE	Object type	***ALL**, *PGM, *LNG	Optional
CHKSIG	Check signature	***SIGNED**, *ALL, *NONE	Optional
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional
SEQNBR	Sequence number	1-16777215, ***END**	Optional
EXPDATE	File expiration date	*Date*, ***PERM**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
SAVF	Save file	*Qualified object name*	Optional
	Qualifier 1: Save file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TGTRLS	Target release	***CURRENT**, *PRV, V5R2M0, V5R3M0, V5R4M0	Optional

Keyword	Description	Choices	Notes
CLEAR	Clear	*NONE, *ALL, *AFTER, *REPLACE	Optional
DTACPR	Data compression	*DEV, *NO, *YES	Optional
LICACPRQD	License acceptance required	*NO, *YES	Optional

<div align="right">Top</div>

Product (LICPGM)

Specifies the seven-character identifier of the licensed program that is saved.

This is a required parameter.

<div align="right">Top</div>

Device (DEV)

Specifies the names of the tape or optical devices used for the save licensed program operation. Each device name must already be known on the system by a device description. Use the Work with Device Descriptions (WRKDEVD) command to display the names of the devices available on this system.

This is a required parameter.

The possible values are:

*SAVF The product is saved in the save file specified on the save file (SAVF) parameter.

optical-device-name
> Specify the name of the optical device to be used for the save operation.

tape-media-library-device-name
> Specify the name of the tape media library device used for the save operation.

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

<div align="right">Top</div>

Optional part to be saved (OPTION)

Specifies the optional parts of the licensed program given in the **Product** prompt (LICPGM parameter) that are saved.

The possible values are:

*BASE
> Only the base part of the licensed program is to be saved.

number-of-licensed-program-option
> Specify the number of the optional part of the listed licensed program that is to be saved.

<div align="right">Top</div>

Release (RLS)

Specifies which version, release, and modification level of the licensed program is saved.

The possible values are:

*ONLY
Only one version, release, and modification level is installed for the licensed program option.

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

Language for licensed program (LNG)

Specifies which national language version (NLV) is used for the save operation.

Note: This parameter is ignored when OBJTYPE(*PGM) is specified.

The possible values are:

*PRIMARY
The primary language is saved. The primary language is the language of the operating system.

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

*ALL   All languages are saved.

feature-code
Specify the NLV identifier for the language that is saved for the licensed program.

Object type (OBJTYPE)

Specifies the type of licensed program objects being saved.

The possible values are:

*ALL   Program and language objects specified on the LNG parameter are saved.

*PGM   Only the program objects for the licensed program are saved.

*LNG   The objects associated with the NLV identified on the LNG parameter are saved.

Check signature (CHKSIG)

Specifies if the digital signatures of objects being saved with the licensed program are to be checked.

The possible values are:

*SIGNED

> Objects with digital signatures are checked. Objects that are not digitally signed will not have their signatures verified. Any signed object with a signature that is not valid will be identified in a message in the job log and the save will fail.

*ALL   All objects that can be digitally signed are checked. Any object that can be signed but has no signature will be identified in a message in the job log. The save will not be terminated for unsigned objects. Any signed object with an invalid digital signature will be identified in a message in the job log and the save will fail.

*NONE

> Digital signatures of objects will not be checked.

Volume identifier (VOL)

Specifies the volume identifiers of the tape volumes on which the licensed program data is saved. The volumes must be placed in the same order as the volume identifiers are specified on this parameter. Up to 75 entries can be specified.

The possible values are:

*MOUNTED

> The volume currently placed in the device is used.

For a media library device, the volume to be used is the next cartridge in the the category mounted by the Set Tape Category (SETTAPCGY) command.

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

volume-identifier

> Specify the identifiers of one or more volumes in the order in which they are placed in the device and used to save the licensed program.

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END   The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215

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

File expiration date (EXPDATE)

Specifies the expiration date of the tape created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

*PERM

> The file is protected permanently.

date   Specify the date when protection for the file ends.

End of media option (ENDOPT)

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

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

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

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

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

Save file (SAVF)

Specifies the qualified name of the existing save file to which the product is saved. The save file must be empty, unless *ALL is specified on the Clear prompt (CLEAR parameter).

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

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

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

The possible values are:

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

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to restore and use the product. In the examples given for the *CURRENT and *PRV values, and when specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V4R1M0 is version 4 release 1 modification level 0.

Note: This value is limited by the minimum operating system release specified when the product was created.

The possible values are:

*CURRENT
>The product is to be restored to, and used on, the release of the operating system currently running on your system. The product can also be restored to a system with any subsequent release of the operating system installed.

*PRV The product is to be restored to the previous release with modification level 0 of the operating system. The product can also be restored to a system with any subsequent release of the operating system installed.

target-release
>Specify the release in the format VxRxMx. The product can be restored to a system with the specified release or with any subsequent release of the operating system installed.

>Valid values depend on the current version, release, and modification level, and they change with each new release. Press F4 to see a list of valid target release values.

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE
>None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*ALL All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

*AFTER
>All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

>**Note:** The *AFTER value is not valid for save files.

*REPLACE
>Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Data compression (DTACPR)

Specifies whether data compression is used.

The possible values are:

*DEV   If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

*NO   No data compression is performed.

*YES   If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to a diskette or save file, software compression is performed. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

License acceptance required (LICACPRQD)

Specifies whether or not the user will be prompted for license acceptance when the saved licensed program is restored. If the licensed program being saved does not have online software agreements, no prompt will be done at restore time. This parameter does not affect the license acceptance status of the licensed program currently installed.

The possible values are:

*NO   License acceptance will not be required when the saved licensed program is restored.

*YES   License acceptance will be required when the saved licensed program is restored, if the licensed program being saved has online software agreements.

Examples

Example 1: Tapes Cleared Automatically

```
SAVLICPGM   LICPGM(5722WDS)   DEV(TAP01)   CLEAR(*ALL)
```

This command saves the *BASE option of the WebSphere Development Studio licensed program (5722WDS). It is saved on the tape that is in the TAP01 tape drive. Each uncleared tape is cleared automatically when it is encountered, and the save operation continues without operator intervention.

Example 2: Saving on Labeled Tape Volume

```
SAVLICPGM   LICPGM(5722WDS)   DEV(TAP01)   VOL(ABCDE)
```

The *BASE option of the WebSphere Development Studio licensed program (5722WDS) is saved on the TAP01 tape drive, starting on the tape volume labeled ABCDE. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be placed in the TAP01 tape drive is shown to the operator.

Example 3: Saving on Multiple Volumes

```
SAVLICPGM   LICPGM(5722SS1)  OPTION(2)  DEV(TAP01 TAP02)
```

The online help option of the i5/OS licensed program (5722SS1) is saved on tape drives TAP01 and TAP02 in alternating order. If the save operation exceeds the storage capacity of two tapes, a message requesting that another volume be put on TAP01 is shown to the operator. The tapes are rewound at the completion of the save operation.

Example 4: Saving to Save Files Cleared Automatically

```
SAVLICPGM   LICPGM(5722WDS)  DEV(*SAVF)
            CLEAR(*ALL)  SAVF(MYLIB/MYSAVF)
```

This command saves the base option of the WebSphere Development Studio licensed program (5722WDS). It is saved to the save file MYSAVF in the library MYLIB. The save file MYSAVF is cleared automatically and the save operation continues without operator intervention.

Example 5: Saving program objects to a save file

```
SAVLICPGM   LICPGM(1MYPROD)  OPTION(*BASE)
            DEV(*SAVF)  RLS(*ONLY)
            OBJTYPE(*PGM)  SAVF(MYLIB/MYSAVF)
```

The 1MYPROD product consists of two libraries and does not have any folders. Only the program objects for 1MYPROD product will be saved in the save file, MYSAVF, in MYLIB library.

Example 6: Saving all objects in one library to a save file

```
SAVLICPGM   LICPGM(1MYPROD)  OPTION(*BASE)
            DEV(*SAVF)  RLS(*ONLY)
            LNG(2924)  OBJTYPE(*ALL)  SAVF(MYLIB/MYSAVF)
```

The 1MYPROD product consists of only one library and does not have any folders. The program objects and language objects for the NLV 2924 of 1MYPROD product will be saved in the save file, MYSAVF, in MYLIB library.

Example 7: Saving all objects in two libraries to a save file

```
SAVLICPGM   LICPGM(1MYPROD)  OPTION(*BASE)
            DEV(*SAVF)  RLS(*ONLY)
            LNG(2924)  OBJTYPE(*ALL)  SAVF(MYLIB/MYSAVF)
```

The 1MYPROD product consists of two libraries and has one root folder with the *PGM objects, and one subfolder of that root folder with the *LNG objects. The program objects and language objects for the NLV 2924 of 1MYPROD product will be saved in the save file, MYSAVF, in MYLIB library.

Example 8: Saving for a previous release.

```
SAVLICPGM   LICPGM(1MYPROD)  DEV(TAP01)  TGTRLS(*PRV)
```

The 1MYPROD product will be saved to TAP01 and may be restored on a system with the previous release of the operating system installed.

Top

Error messages

<u>*ESCAPE Messages</u>

CPF37A2

   Installation cannot be performed.

CPF3728

   Device &1 specified with other devices.

CPF3733

   &2 &1 in &3 previously damaged.

CPF3884

   Licensed program &1 option &2 not processed.

Top

Save Object (SAVOBJ)

The Save Object (SAVOBJ) command saves a copy of a single object or a group of objects located in the same library. When *ALL is specified for the **Objects (OBJ)** parameter, objects can be saved from a list of libraries. When saving to a save file, only one library can be specified. The system saves the specified objects by writing a copy of each object on tape or optical media, or in a save file. The objects are not affected in the system unless the command specifies that the storage should be freed. However, the description of each object is changed with the date, time, and place when it was last saved, unless *NO is specified for the **Update history (UPDHST)** parameter.

For job queues, user queues, message queues, and logical files, only the object descriptions are saved, and the contents of the objects are not saved. However, logical file access paths can be saved by specifying *YES for the **Save access paths (ACCPTH)** parameter. The contents of spooled files on output queues can be saved by specifying *ALL for the **Spooled file data (SPLFDTA)** parameter. The contents of a save file can be saved by specifying *YES for the **Save file data (SAVFDTA)** parameter or using the Save Save File Data (SAVSAVFDTA) command. The contents of a data queue can be saved by specifying *DTAQ for the **Queue data (QDTA)** parameter.

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

Restrictions:
- You must either have save system (*SAVSYS) special authority or the following object authorities:
  - object existence (*OBJEXIST) authority for each object to be saved (for spooled file data, *OBJEXIST authority for the output queue)
  - execute (*EXECUTE) authority to each library objects are saved from
- If you do not have *SAVSYS special authority, only those objects that you have authority for are saved.
- When saving to a tape or optical device, you must have use (*USE) authority to the device description and device file. When saving to a save file, you must have object operational (*OBJOPR) and add (*ADD) authorities to the save file, and *EXECUTE authority to the library where the save file is located.
- When using a media definition, you must have *USE authority to the media definition and *EXECUTE authority to the media definition library.
- This command does not save the data dictionary for the library or its associated database files. To save them, the Save Library (SAVLIB) command should be used.
- When using the OUTFILE parameter to save to an existing database file, you must have *EXECUTE authority to the output file library.
- If tape is used, a standard-labeled volume must be placed in the device.
- No object being saved can be changed by a job that is running at the time the save operation occurs unless save-while-active is used.
- When the contents of a save file are being saved to the same save file by specifying SAVFDTA(*YES), only the description of the save file is saved.
- When the contents of a save file are saved by specifying SAVFDTA(*YES), the save file must be restored before objects contained in it can be restored.
- When using the **Command user space (CMDUSRSPC)** parameter, you must have *USE authority to the user space and *EXECUTE authority to the user space library.

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 1
LIB	Library	Single values: *USRSPC Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 2
DEV	Device	Single values: *SAVF, *MEDDFN Other values (up to 4 repetitions): *Name*	Required, Positional 3
OBJTYPE	Object types	Single values: ***ALL** Other values (up to 300 repetitions): *Character value*	Optional, Positional 4
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional, Positional 5
SEQNBR	Sequence number	1-16777215, ***END**	Optional
LABEL	Label	*Character value*, ***LIB**	Optional
EXPDATE	File expiration date	*Date*, ***PERM**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
SAVF	Save file	*Qualified object name*	Optional
	Qualifier 1: Save file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MEDDFN	Media definition	*Qualified object name*	Optional
	Qualifier 1: Media definition	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OPTFILE	Optical file	*Path name*, *****	Optional
USEOPTBLK	Use optimum block	***YES**, *NO	Optional
TGTRLS	Target release	***CURRENT**, *PRV, V5R2M0, V5R3M0, V5R4M0	Optional
UPDHST	Update history	***YES**, *NO	Optional
CLEAR	Clear	***NONE**, *ALL, *AFTER, *REPLACE	Optional
PRECHK	Object pre-check	***NO**, *YES	Optional
SAVACT	Save active	***NO**, *LIB, *SYNCLIB, *SYSDFN	Optional
SAVACTWAIT	Save active wait time	*Element list*	Optional
	Element 1: Object locks	0-99999, **120**, *NOMAX	
	Element 2: Pending record changes	0-99999, ***LOCKWAIT**, *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, ***LOCKWAIT**, *NOMAX	
SAVACTMSGQ	Save active message queue	*Qualified object name*	Optional
	Qualifier 1: Save active message queue	*Name*, ***NONE**, *WRKSTN	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
FILEMBR	File member	Values (up to 50 repetitions): *Element list*	Optional
	Element 1: File	*Name*, ***ALL**	
	Element 2: Member	Single values: ***ALL**, *NONE Other values (up to 50 repetitions): *Generic name, name*	
ACCPTH	Save access paths	***SYSVAL**, *NO, *YES	Optional
SAVFDTA	Save file data	***YES**, *NO	Optional
SPLFDTA	Spooled file data	***NONE**, *ALL	Optional

Keyword	Description	Choices	Notes
QDTA	Queue data	***NONE**, *DTAQ	Optional
STG	Storage	***KEEP**, *FREE	Optional
DTACPR	Data compression	***DEV**, *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	***DEV**, *NO	Optional
OMITLIB	Libraries to omit	Single values: ***NONE**, *USRSPC Other values (up to 300 repetitions): *Generic name, name*	Optional
OMITOBJ	Objects to omit	Single values: *USRSPC Other values (up to 300 repetitions): *Element list*	Optional
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Generic name, name*, ***NONE**, *ALL	
	Qualifier 2: Library	*Generic name, name*, ***ALL**	
	Element 2: Object type	*Character value*, ***ALL**	
ASPDEV	ASP device	*Name*, *****, *SYSBAS, *CURASPGRP	Optional
OUTPUT	Output	***NONE**, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	*Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	
INFTYPE	Type of output information	***OBJ**, *LIB, *MBR, *ERR	Optional
CMDUSRSPC	Command user space	*Qualified object name*	Optional
	Qualifier 1: Command user space	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Objects (OBJ)

Specifies the names of one or more objects or the generic name of each group of objects to be saved. All the objects must be in the library specified for the **Library (LIB)** parameter. If *ALL is specified or defaulted for the **Object types (OBJTYPE)** parameter, all the object types listed in the description of that parameter are saved, provided they are in the specified library and have the specified names.

This is a required parameter.

Single values

*ALL   All the objects in the specified libraries are saved, depending on the values specified for the OBJTYPE parameter.

Other values (up to 300 repetitions)

generic-name
        Specify one or more generic names of groups of objects in the specified library to be saved. A

generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete object name.

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

Library (LIB)

Specifies which libraries contain the objects to be saved. If *ALL is specified for the **Objects (OBJ)** parameter, up to 300 library names can be specified.

This is a required parameter.

Single values

*USRSPC
> The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are saved. Only one library can be specified in the user space when saving to a save file. Special value *SPLF cannot be specified in the user space.

Other values (up to 300 repetitions)

generic-name
> Specify the generic name of the libraries that contain objects to be saved. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name.
>
> **Note:** A generic library name cannot be specified when saving to a save file.

name Specify the name of the library that contains objects to be saved.

> **Note:** Only one library can be specified when saving to a save file.

Device (DEV)

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

This is a required parameter.

Single values

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

*MEDDFN
> The save operation is done using the devices and media identified in the media definition specified for the **Media definition (MEDDFN)** parameter.

Other values

optical-device-name
> Specify the name of the optical device used for the save operation.

tape-media-library-device-name
> Specify the name of the tape media library device used for the save operation.

tape-device-name
> Specify the names of one or more tape devices used for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Object types (OBJTYPE)

Specifies the types of system objects to be saved.

Single values

*ALL   All object types that are specified by name and are in the specified library are saved. If *ALL is also specified for the **Objects (OBJ)** parameter, all the objects in the library that are of the types that can be saved are saved.

Other values (up to 300 repetitions)

object-type
> Specify the value for each of the types of objects to be saved, such as command (*CMD), file (*FILE), or program (*PGM).

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

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED
> The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

Other values (up to 75 repetitions)

character-value
> Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215

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

Label (LABEL)

Specifies the name that identifies the data file on the tape volume that is to be used for the save operation. If this parameter is used on the save command, the same label must be specified on the restore command.

Note: You cannot specify *SAVLIB on this parameter, since it is a special value for the **Label (LABEL)** parameter of the restore command and would prevent you from restoring what you saved.

*LIB The file label is created by the system using the name of the library specified for the **Library (LIB)** parameter.

character-value

  Specify the data file identifier of the data file used for the save operation. A maximum of 17 characters can be used. This option is valid only for a single-library save operation.

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

*PERM

  The file is protected permanently.

date Specify the date when protection for the file ends.

End of media option (ENDOPT)

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

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

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

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

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

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

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

Qualifier 1: Save file

name   Specify the name of save file to be used.

Qualifier 2: Library

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

*CURLIB
> The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

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

Media definition (MEDDFN)

Specifies the media definition (*MEDDFN) object that identifies the devices and media used to contain the saved data. For information about creating and using a media definition, see the Backup and Recovery book, SC41-5304, and the Create Media Definition API in the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

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

Qualifier 1: Media definition

name   Specify the name of the media definition to be used.

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Optical file (OPTFILE)

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

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

*   The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/'*
    The system generates an optical file name in the specified directory of the optical volume.

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

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

*YES   The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

*NO   The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to restore and use the object.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

To specify that an object be saved for distribution to a system at a different release level than the system on which the save operation is to occur, the procedure differs for program or non-program objects and by

the release level on which a program object is created. If, for example, you are saving an object for distribution to a target system running on an earlier release, you have the following choices:

For program objects:
- If the program object was created at a release level more current than the targeted earlier release, you must:
    1. create the program object again specifying the targeted earlier release
    2. save the program object specifying the targeted earlier release
    3. restore the program object on the target system.
- If the program object was created at the same release level as the target system, you can:
    1. save the program object specifying the targeted earlier release
    2. restore the program object on the target system.

For non-program objects:

You can:
1. save the object specifying the targeted earlier release
2. restore the object on the target system.

*CURRENT
> The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value
> Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Update history (UPDHST)

Specifies whether the save history information of each saved object is changed with the date, time, and location of this save operation. The save history information for an object is displayed using the Display Object Description (DSPOBJD) command. The save history information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) or FROMENTLRG(*LASTSAVE) are used on the Apply Journaled Changes (APYJRNCHG) command.

*YES The last save date, time, and location is updated in each object saved.

*NO The save history information contained in the description of each object saved is not updated.

> **Note:** UPDHST(*NO) should be used for a save operation that is not intended for recovery. For example, if the save data is sent, record by record, to another system and the save file immediately deleted, the save history information is probably not to be updated.

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.

2. Clearing an optical volume does initialize it.

3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE
> None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*ALL   All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

*AFTER
> All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

> **Note:** The *AFTER value is not valid for save files.

*REPLACE
> Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:

1. The objects do not exist

2. The library or the objects were previously found to be damaged

3. The library or the objects are locked by another job

4. The requester of the save operation does not have authority for the library or to save the objects.

*NO   The save operation for a library continues, saving only those objects that can be saved.

*YES   If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation

continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are
specified and an object in any library to be saved does not meet the preliminary check conditions,
the save operation ends and no objects are saved.

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is
performed as if SAVACT(*NO) was specified.

*NO   Objects that are in use are not saved. Objects cannot be updated while being saved.

*LIB   Objects in a library can be saved while they are in use by another job. All of the objects in a
library reach a checkpoint together and are saved in a consistent state in relationship to each
other.

*SYNCLIB

> Objects in a library can be saved while they are in use by another job. All of the objects and all of
> the libraries in the save operation reach a checkpoint together and are saved in a consistent state
> in relationship to each other.
>
> **Note:** If you specify this value and you are saving many libraries, it can take a long time to reach
> a checkpoint for all of the objects and libraries in the save operation.

*SYSDFN

> Objects in a library can be saved while they are in use by another job. Objects in a library may
> reach checkpoints at different times and may not be in a consistent state in relationship to each
> other.

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes
to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an
object remains in use for the specified time, the object is not saved.

120   The system waits up to 120 seconds for each individual object lock before continuing the save
operation.

*NOMAX

> No maximum wait time exists.

0-99999

> Specify the number of seconds to wait for each individual object lock before continuing the save
> operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for
transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)**
parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved

must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value *NOCMTBDY is specified.

*LOCKWAIT
>The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

*NOCMTBDY
>The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

>If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify *NOCMTBDY.

>**Note:** This value cannot be specified if the **Target release (TGTRLS)** parameter value is earlier than V5R3M0.

*NOMAX
>No maximum wait time exists.

0-99999
>Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.
- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

*LOCKWAIT
>The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

*NOMAX
>No maximum wait time exists.

0-99999
>Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.

>If 0 is specified, and only one name is specified for the **Objects (OBJ)** parameter, and *FILE is the only value specified for the **Object types (OBJTYPE)** parameter, the system will save the object without requiring the types of transactions that are listed above to reach a commit boundary.

Top

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE
No notification message is sent.

*WRKSTN
The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name   Specify the name of the message queue to be used.

Qualifier 2: Library

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

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

name   Specify the name of the library where the message queue is located.

File member (FILEMBR)

Specifies the database file members that are saved. This parameter is made up of two parts: the file name and the member name.

Each database file specified here must also be specified for the **Objects (OBJ)** parameter, by its complete name, a generic name, or *ALL. The **Object types (OBJTYPE)** parameter value must be *ALL or include *FILE.

Note: This parameter cannot be specified when STG(*FREE) is specified.

Element 1: File

*ALL   The list of member name values that follows this value applies to all files specified for the OBJ parameter.

name   Specify the name of the database file from which the listed members are to be saved. Up to 50 files can be specified, with a member list for each file.

> **Note:** Generic names are not valid for the database file name, but are allowed for the member name.

> **Note:** Duplicate file names are not allowed.

Element 2: Member

Single values

*ALL   All members are saved from the specified file.

*NONE

No members are saved from the specified file. Only the file description is saved.

Other values (up to 50 repetitions)

generic-name

Specify the generic names of the members to be saved from the specified file. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete member name.

Note: If generic member names are specified, the file must contain member names that match the generic names for the file to be saved. For example, if PAY* is specified as a generic member name, and the system is unable to find a member whose name starts with PAY, the file is not saved. If files specified by the FILEMBR parameter are not saved because members with the specified generic name cannot be found, a diagnostic message is sent, the save operation ends, and an escape message is sent specifying the number of files not saved. If at least one of the files processed for the FILEMBR parameter contains a member with the specified generic name, the diagnostic message is not sent, and the number of files not saved is in the final completion message.

name   Specify the names of the members to be saved from the given file.

Note: If specific member names are specified, the specified members must exist in the file for any part of the file to be saved or restored.

Save access paths (ACCPTH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Backup and Recovery book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

*SYSVAL

The QSAVACCPTH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

*NO   Only those objects specified on the command are saved. No logical file access paths are saved.

*YES   The specified physical files and all eligible logical file access paths over them are saved.

　　　Note: Specifying this value does not save the logical files.

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

*YES   The description and contents of a save file are saved.

*NO   Only the description of a save file is saved.

Spooled file data (SPLFDTA)

Specifies whether to save spooled file data and attributes for output queues that are saved.

*NONE
　　　No spooled file data is saved.

*ALL   For each output queue that is saved, all available spooled file data on the output queue is saved.

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

*NONE
　　　Only the description of a queue is saved.

*DTAQ
　　　The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Storage (STG)

Specifies whether the system storage that is occupied by the data portion of the specified members (except for save files), modules, programs, service programs, Structured Query Language (SQL) packages, and journal receivers in the library being saved is freed as part of the save operation. Only the data portion of the objects is freed, not the descriptions of the objects.

*KEEP
　　　The storage occupied by the data portion of the objects being saved is not freed.

*FREE   The storage occupied by the data portion of the specified objects being saved is freed as part of the save operation. The storage for all the objects in a library is freed only after all the objects in that library are saved successfully.

Note: To prevent the possible abnormal end of a program, the program being saved must not be running in the system when *FREE is specified.

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

*DEV If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

*NO No data compression is performed.

*YES If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

*LOW If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

   Note: This value is not valid for tape.

*MEDIUM
   If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

   Note: This value is not valid for tape.

*HIGH
   If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

   Note: This value is not valid for tape.

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

*DEV Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

   Note: If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

*NO   Device data compaction is not performed.

Libraries to omit (OMITLIB)

Specifies the names of one of more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

*NONE
>   No libraries are excluded from the save operation.

*USRSPC
>   The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

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

name   Specify the name of the library to be excluded from the save operation.

Objects to omit (OMITOBJ)

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

Single values

*USRSPC
>   The objects identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

Element 1: Object

Qualifier 1: Object

*NONE
>   No objects are excluded from the operation.

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

generic-name
>   Specify the generic name of the objects to be excluded.

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

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

Qualifier 2: Library

<u>*ALL</u>   The specified objects are excluded from all libraries that are part of the operation.

generic-name
Specify the generic name of the libraries that contain objects to be excluded.

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

name   Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

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

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

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

<div align="right">Top</div>

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

<u>*</u>   The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS
The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP
If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name   Specify the name of the ASP device to be included in the save operation.

<div align="right">Top</div>

Output (OUTPUT)

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

*NONE
>   No output listing is created.

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

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

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

File to receive output (OUTFILE)

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

Qualifier 1: File to receive output

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

Qualifier 2: Library

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

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

name   Specify the name of the library to be searched.

Output member options (OUTMBR)

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

Element 1: Member to receive output

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

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

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

Element 2: Replace or add records

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

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

Top

Type of information (INFTYPE)

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

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

*ERR The list contains information about the command, an entry for each library, and an entry for each object that was not successfully saved.

*LIB The list contains a library entry for each library requested to be saved

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

Top

Command user space (CMDUSRSPC)

Specifies a user space containing the values for the parameters which have *USRSPC specified for this command. The user space allows up to 32767 list values for each parameter, while the command parameters only allow up to 300 list values. The user space must define the parameters in the format used by the Save Object List (QSRSAVO) API.

Qualifier 1: User space

name Specify the name of the user space containing the values for the parameters which have *USRSPC specified for this command.

Qualifier 2: Library

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

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

name Specify the name of the library where the user space is located.

Top

Examples

Example 1: Saving Program and File With Same Name

```
SAVOBJ   OBJ(PETE)   LIB(LIBX)   DEV(TAP01)
```

This command saves the objects named PETE which are located in the LIBX library. If, for example, LIBX contains both a program and a file named PETE, both objects are saved. The storage occupied by the object is not freed because the STG parameter default (*KEEP) was assumed.

Example 2: Freeing System Storage

```
SAVOBJ   OBJ(MSTRPAY PAY*)   LIB(QGPL)   DEV(TAP01)   STG(*FREE)
```

The object named MSTRPAY, and all the objects whose names start with the characters PAY located in the general purpose library (QGPL), are saved. As part of the save operation, the system storage that was occupied by the data portion of the saved member, module, program, service program, SQL package, and journal receiver objects is freed.

Example 3: Saving File on Optical

```
SAVOBJ   OBJ(FILEA)  OBJTYPE(*FILE)  LIB(LIBY)  DEV(OPT01)
         VOL(TOM)  CLEAR(*REPLACE)
```

The file named FILEA in the LIBY library is saved in a file with the library name LIBY on the optical volume that is identified by the volume identifier TOM. If the LIBY file already exists on the optical volume, it is automatically replaced when FILEA is saved.

Example 4: Saving Objects Supported on Previous Release

```
SAVOBJ   OBJ(PAY*)  LIB(LIB1)  DEV(TAP01)  TGTRLS(*PRV)
```

This command saves the objects beginning with the characters PAY from the LIB1 library in a format compatible with the previous release of the operating system. Only those objects supported on the previous release are saved.

Example 5: Saving Description and Data of File

```
SAVOBJ   OBJ(SAVEFILE)  LIB(MYLIB)  OBJTYPE(*FILE)  DEV(TAP01)
         SAVFDTA(*YES)
```

This command saves the file named SAVEFILE which is located in the library named MYLIB. Both the description and the data are saved for this save file.

Error messages

*ESCAPE Messages

CPF3701
> &1 objects saved from &3. &2 not saved.

CPF3702
> &1 objects saved from &3. &2 not saved. &9 not included.

CPF3703
> &2 &1 in &3 not saved.

CPF3708
> Save file &1 in &2 too small.

CPF3709
> Tape devices do not support same densities.

CPF3727
> Duplicate device &1 specified on device name list.

CPF3728
> Device &1 specified with other devices.

CPF3730
> Not authorized to &2 &1 in library &3.

CPF3731
> Cannot use &2 &1 in library &3.

CPF3733

&2 &1 in &3 previously damaged.

CPF3735

Storage limit exceeded for user profile &1.

CPF3738

Device &1 used for save or restore is damaged.

CPF3747

Object names cannot be specified with more than one library.

CPF3749

Objects from library &2 not saved.

CPF3767

Device &1 not found.

CPF3768

Device &1 not valid for command.

CPF377D

Save ended because of read error on internal system resource.

CPF377E

Not enough storage for save-while-active request.

CPF377F

Save-while-active request prevented by pending record changes.

CPF3770

No objects saved or restored for library &1.

CPF3771

&1 objects saved from &3. &2 not saved.

CPF3774

&1 objects saved from &3. &2 not saved. &8 not included.

CPF3778

Not all objects saved from all libraries.

CPF378A

Message queue not available.

CPF378C

SAVACTMSGQ(*WRKSTN) not valid for batch job.

CPF378E

Library &1 not saved.

CPF3781

Library &1 not found.

CPF3782

File &1 in &2 not a save file.

CPF3789

Only one library allowed with specified parameters.

CPF379E

Not enough storage available to save library &1.

CPF3793

Machine or ASP storage limit reached.

CPF3794

Save or restore operation ended unsuccessfully.

CPF3797

Objects from library &3 not saved. Save limit exceeded.

CPF37AB

*NOCMTBDY not allowed with target release.

CPF37AC

Library not allowed with *NOCMTBDY.

CPF37B1

SPLFDTA not allowed with target release.

CPF37B4

User space &1 in &2 not valid.

CPF380B

Save cannot be completed at this time.

CPF3812

Save file &1 in &2 in use.

CPF3815

Save file &1 in &2 too small for save operation.

CPF384E

USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3867

Contents of FILEMBR parameter not correct.

CPF3868

FILEMBR specified but OBJTYPE must be *ALL or *FILE.

CPF3871

No objects saved or restored; &3 objects not included.

CPF388B

Optical file path name not valid.

CPF3892

&2 &1 in &3 not saved.

CPF3894

Cancel reply received for message &1.

CPF38A2

ASP device &1 not correct.

CPF38A3

File &1 in &2 not valid with ASPDEV.

CPF38A4

ASP device &1 not correct.

CPF5729

Not able to allocate object &1.

CPF9809

Library &1 cannot be accessed.

CPF9812

File &1 in library &2 not found.

CPF9814

        Device &1 not found.

CPF9820

        Not authorized to use library &1.

CPF9822

        Not authorized to file &1 in library &2.

CPF9825

        Not authorized to device &1.

CPF9833

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

CPFB8ED

        Device description &1 not correct for operation.

OPT1332

        Optical volume not found in device &1.

<u>***STATUS Messages**</u>

CPF3770

        No objects saved or restored for library &1.

CPF3771

        &1 objects saved from &3. &2 not saved.

CPF3871

        No objects saved or restored; &3 objects not included.

Top

Save Restore (SAVRST)

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

The Save/Restore IFS (SAVRST) command saves and restores a copy of one or more objects, that can be used in the integrated file system (IFS).

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

The SAVRST command uses the current save and restore support for objects in libraries and for document library objects. As a result, there are restrictions when you use the SAVRST command on these objects.

Restrictions:

1. For name patterns in the root directory:
   a. OBJ must be one of the following:
      - OBJ(('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT))
      - OBJ(('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT) ('/other values' *OMIT)
2. For names involving objects in libraries:
   a. OBJ must have only one name, except when omitting /QSYS.LIB from /*
   b. OBJ must be one of the following:
      - OBJ('/QSYS.LIB/libname.LIB')
      - OBJ('/QSYS.LIB/libname.LIB/*')
      - OBJ('/QSYS.LIB/libname.LIB/*.type')
      - OBJ('/QSYS.LIB/libname.LIB/objname.type')
      - OBJ('/QSYS.LIB/libname.LIB/filename.FILE/*')
      - OBJ('/QSYS.LIB/libname.LIB/filename. FILE/*.MBR')
      - OBJ('/QSYS.LIB/libname.LIB/filename. FILE/membername.MBR')
      - OBJ('/QSYS.LIB/*.type')
      - OBJ('/QSYS.LIB/objname.type')
      - OBJ('/QSYS.LIB/filename.FILE/*')
      - OBJ('/QSYS.LIB/filename.FILE/*.MBR')
      - OBJ('/QSYS.LIB/filename. FILE/membername.MBR')
   c. The .type must be an object type supported by SAVOBJ and RSTOBJ
   d. libname cannot be QSYS, QDOC, QDOCxxxx, QTEMP, QSPL, QSPLxxxx, QSRV, QRECOVERY, QRPLOBJ, or QSR if libname.LIB is the last component of the name
   e. SUBTREE must be *ALL
   f. For SAVRST:
      - CHGPERIOD end date and end time must be *ALL
      - CHGPERIOD must be default if a file member is specified
      - An object cannot be renamed
      - For database file members, OPTION(*NEW) only restores members for new files
3. For names involving document library objects:

a. OBJ must have only one name, except when omitting /QDLS from /*

b. OBJ and SUBTREE must be one of the following:
   - OBJ('/QDLS/path/foldername') SUBTREE(*ALL)
   - OBJ('/QDLS/path/documentname') SUBTREE(*OBJ)

c. For SAVRST:
   - The defaults must be taken on the PRECHK and SAVACTMSGQ parameters
   - CHGPERIOD must be default with OBJ('/QDLS/path/documentname') SUBTREE(*OBJ)
   - CHGPERIOD start date cannot be *LASTSAVE
   - CHGPERIOD end date and end time must be *ALL
   - SAVACT cannot be *SYNC
   - SAVACTMSGQ must be *NONE
   - ALWOBJDIF must be *NONE or *ALL
   - OPTION must be *ALL

4. Both systems intended to participate in the save and restore operation must be connected to the same APPN network, or, if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Parameters

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location name	*Name*	Required, Positional 1
OBJ	Objects	Values (up to 50 repetitions): *Element list*	Optional, Positional 2
	Element 1: Name	*Path name*, *****	
	Element 2: Include or omit	***INCLUDE**, *OMIT	
PATTERN	Name pattern	Values (up to 50 repetitions): *Element list*	Optional
	Element 1: Pattern	*Character value*, *****	
	Element 2: Include or omit	***INCLUDE**, *OMIT	
SUBTREE	Directory subtree	***ALL**, *DIR, *OBJ, *NONE	Optional
CHGPERIOD	Time period for last change	*Element list*	Optional
	Element 1: Start date	*Date*, ***ALL**	
	Element 2: Start time	*Time*, ***ALL**	
	Element 3: End date	*Date*, ***ALL**	
	Element 4: End time	*Time*, ***ALL**	
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
PRECHK	Object pre-check	***NO**, *YES	Optional
SAVACT	Save active	***NO**, *YES, *SYNC	Optional
SAVACTMSGQ	Save active message queue	*Path name*, ***NONE**, *WRKSTN	Optional
ASPDEV	ASP device	*Name*, ***ALLAVL**, *, *SYSBAS, *CURASPGRP	Optional
OPTION	Option	***ALL**, *NEW, *OLD	Optional
ALWOBJDIF	Allow object differences	Single values: ***NONE**, *ALL Other values (up to 2 repetitions): *OWNER, *AUTL, *PGP	Optional

Keyword	Description	Choices	Notes
FRCOBJCVN	Force object conversion	Single values: ***SYSVAL**, *NO Other values: *Element list*	Optional
	Element 1: Convert during restore	***YES**	
	Element 2: Objects to convert	***RQD**, *ALL	
SCAN	Scan objects	*Element list*	Optional
	Element 1: Scan during save	***NO**, *YES	
	Element 2: Save failed objects	***NOSAVFAILED**, *SAVFAILED	
CRTPRNDIR	Create parent directories	***NO**, *YES	Optional
PRNDIROWN	Parent directory owner	*Simple name*, ***PARENT**	Optional

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format ccccccccc or nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name.

remote-location-name
> Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name
> Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Objects (OBJ)

Specifies the objects to be saved. You can specify an object name pattern for the path name to be used. When a path name is specified that could match many objects, you can specify a value for the **Name pattern (PATTERN)** parameter to subset the objects that are to be saved.

A maximum of 300 path names can be specified.

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

Additional information about object name patterns is in the Integrated file system information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Element 1: Name

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

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

Element 2: Include or omit

Specifies whether names that match the pattern should be included or omitted from the operation. Note that in determining whether a name matches a pattern, relative name patterns are always treated as relative to the current working directory.

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

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

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

Note: The objects will be restored with the same name.

Name pattern (PATTERN)

Specifies one or more object name patterns to be used to subset the objects to be saved. The **Objects (OBJ)** parameter defines the set of candidate objects. A maximum of 50 values can be specified for this parameter.

Element 1: Pattern

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

character-value
>Specify an object name or a pattern that can match many names.

Element 2: Include or omit

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

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

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

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

Directory subtree (SUBTREE)

Specifies whether directory subtrees are included in the save operation.

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

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

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

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

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

Top

Time period for last change (CHGPERIOD)

Specifies a date/time range. Objects that were last changed within that range will be saved.

Element 1: Start date

*ALL   No starting date is specified. All objects last changed prior to the ending date will be saved.

*LASTSAVE
The objects that have changed since the last time they were saved with UPDHST(*YES) specified are saved. **Notes:**
1. If this value is specified, the value *ALL must be specified for all other elements of this parameter.
2. For local file systems, the system archive attribute is used. For remote file systems, the PC archive attribute is used.

date   Specify the date after which objects that have changed are to be saved. The date must be specified in job date format.

Element 2: Start time

*ALL   All times of day are included in the range.

time   Specify the time on the start date after which objects that have changed are to be saved.

The time is specified in 24-hour format with or without a time separator as follows:
- With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

Note: Specifying an explicit time is valid only if the starting date is an explicit date.

Element 3: End date

*ALL   No ending date is specified. All objects changed since the starting date will be saved.

date   Specify the date before which objects that have changed are to be saved. The date must be specified in the job date format.

Element 4: End time

*ALL   All times of day are included in the range.

time Specify a time on the end date before which objects that have changed are to be saved.

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

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

Note: Specifying an explicit time is valid only if the ending date is an explicit date.

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

*CURRENT
    The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value
    Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Object pre-check (PRECHK)

Specifies whether the save operation ends if any of the selected objects cannot be saved.

*NO The save operation does not end. Objects that can be saved are saved.

*YES The save operation ends. Nothing is saved unless all of the selected objects can be saved.

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state, this parameter is ignored and the save operation is performed as if SAVACT(*NO) was specified.

*NO Objects that are in use are not saved. Objects cannot be updated while being saved.

*YES Objects can be saved and used at the same time. The object checkpoints can occur at different times.

*SYNC

Objects can be saved and used at the same time. All of the object checkpoints occur at the same time.

Top

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing is complete.

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

*NONE

No notification message is sent.

*WRKSTN

The notification message is sent to the work station message queue.

path-name

Specify the path name of the message queue to be used.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation.

*DFT The operation uses the ASPDEV value appropriate for the file system from which objects are being saved. For Integrated File System objects, *ALLAVL is used. For objects from the QSYS file system, the corresponding save command ASPDEV default is used.

*ALLAVL

The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and all available independent ASPs.

* The operation includes the system ASP, all basic user ASPs, and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS

The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP

If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Top

Option (OPTION)

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

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

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

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

Allow object differences (ALWOBJDIF)

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

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

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

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

Single values

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

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

Other values (up to 3 repetitions)

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

> If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not

exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

*OWNER

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

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

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

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

<div align="right">Top</div>

Force object conversion (FRCOBJCVN)

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

Notes:

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

Single values

*SYSVAL

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

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

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

Element 1: Convert during restore

*YES The objects are converted during the restore operation.

Notes:

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

Element 2: Objects to convert

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

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

Scan objects (SCAN)

Specifies whether objects will be scanned while being saved when exit programs are registered with any of the integrated file system scan-related exit points and whether objects that previously failed a scan should be saved.

The integrated file system scan-related exit points are:
- QIBM_QP0L_SCAN_OPEN - Integrated File System Scan on Open Exit Program
- QIBM_QP0L_SCAN_CLOSE - Integrated File System Scan on Close Exit Program

For details on these exit points, see the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Element 1: Scan during save

*NO   Objects will not be scanned by the scan-related exit programs.

*YES   Objects will be scanned according to the rules described in the scan-related exit programs.

Element 2: Save failed objects

*NOSAVFAILED
            Objects that have either previously failed a scan or that fail a scan by a QIBM_QP0L_SCAN_OPEN exit program during this save will not be saved.

*SAVFAILED
            Objects that have either previously failed a scan or that fail a scan during this save will be saved.

Create parent directories (CRTPRNDIR)

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

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

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

Parent directory owner (PRNDIROWN)

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

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

*PARENT
>   The owner of a parent directory being created by the restore will be the same as the owner of the directory it is being created into. For example, if object '/a/b/c/file1' is being restored and directory '/a' exists but the '/b' and '/b/c' directories do not exist, the '/b' and '/b/c' directories are created with the same owner as the '/a' directory.

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

Top

Examples

Example 1: Saving and Restoring a Member Object

```
SAVRST   RMTLOCNAME(SYSTEM1)
         OBJ(('QSYS.LIB/JTEMP.LIB/ZXC.FILE/QYYCPDGT.MBR'))
```

This command saves the QYYCPDGT member from file ZXC in library JTEMP and restores the object on the iSeries system at remote location SYSTEM1.

Example 2: Saving and Restoring a Directory

```
SAVRST   RMTLOCNAME(SYSTEM2)  OBJ(('MYDIR'))  SAVACT(*YES)
         SAVACTMSGQ('QSYS.LIB/SVRTEST.LIB/ZXC.MSGQ')
```

This command saves the MYDIR directory while active, and will use the ZXC message queue in library SVRTEST to save messages.

Top

Error messages

*ESCAPE Messages

CPCAD80
>   &1 objects saved and restored.

CPFAD8D
>   An error occurred during the &1 operation.

CPFAD80
>   Unable to establish connection from &1 to &2.

CPFAD81
>   User profile &1 not found on remote location &2.

CPFAD82
>   Remote location &1 not found.

CPFAD83
>   Remote location &1 cannot be source location.

CPFAD84

ObjectConnect internal error, function code &1, return code &2.

CPFAD86

Location name &1 unable to close &2.

CPFAD88

Unable to establish connection from &1 to &2.

CPFAD93

APPC failure. Failure code is &3.

CPF389C

ObjectConnect internal error, function code &1, return code &2.

Top

Save Restore Configuration (SAVRSTCFG)

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

The Save/Restore Configuration (SAVRSTCFG) command saves and restores configuration information without requiring a dedicated system. The information saved and restored includes the following:

- Configuration lists (*CFGL)
- Connection lists (*CNNL)
- Class-of-service descriptions (*COSD)
- Internet Packet Exchange descriptions (*IPXD)
- Mode descriptions (*MODD)
- NetBIOS descriptions (*NTBD)

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

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

Restrictions:

1. To use this command, the user must have *SAVSYS authority, or object existence authority for (or be the owner of) each object specified if the object already exists on the system.
2. This command is shipped with public *EXCLUDE authority.
3. The user must have either *ALLOBJ authority or authority to the command.
4. Changes made to the configuration while the SAVRSTCFG command is being run may not be reflected on the restore system, depending on when the changes occurred in relation to the send operation. If the operation cannot obtain a lock on a configuration object, a diagnostic message is issued; and the object is not saved and restored. The operation does not obtain the lock on a configuration object when a change or delete operation against the object does not complete before the default wait timer expires.
5. If any system resource manager objects are in use, a message is issued and none of them are saved and restored. The commands which can cause this to happen are Work with Hardware Products (WRKHDWPRD) and Work with Hardware Resources (WRKHDWRSC). No display commands can cause the system resource manager database to lock. All other known functions do read-only operations.
6. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL, *SRM Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 1

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location name	*Name*	Required, Positional 2
OBJTYPE	Object types	Single values: ***ALL** Other values (up to 5 repetitions): *CFGL, *CNNL, *COSD, *MODD, *NTBD	Optional
SRM	System Resource Management	***NONE**, *TRA	Optional
PRECHK	Object pre-check	***NO**, *YES	Optional
ALWOBJDIF	Allow object differences	Single values: ***NONE**, *ALL Other values (up to 3 repetitions): *AUTL, *OWNER, *PGP	Optional

Objects (OBJ)

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

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

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

This is a required parameter.

Single values

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

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

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

Other values (up to 300 repetitions)

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

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

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

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format ccccccccc or nnnnnnnn.ccccccccc, where nnnnnnnn is the network identifier (ID) and ccccccccc is the remote location name.

remote-location-name
> Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name
> Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Object types (OBJTYPE)

Specifies the types of i5/OS system objects that are saved and restored.

*ALL All configuration object types that are specified by name are saved and restored. If OBJ(*ALL) is specified, all of the saved configuration objects are saved and restored. restored. The following types can be specified:

*CFGL
> All configuration list object types are saved and restored.

*CNNL
> All connection list object types are saved and restored.

*COSD
> All class-of-service description object types are saved and restored.

*IPXD All Internetwork Packet Exchange object types are saved and restored.

*MODD
> All mode description object types are saved and restored.

*NTBD
> All NetBIOS description object types are saved and restored.

System Resource Management (SRM)

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

*NONE
> No SRM information is restored.

*TRA All token-ring adapter information is restored.

Object pre-check (PRECHK)

Specifies whether the save configuration operation ends if any of the objects satisfy the following conditions:

- The objects were previously found to be damaged.
- The objects are locked by another job.
- The user does not have authority to save the objects.

*NO   The save operation continues, saving only configuration and system resource management (SRM) objects that can be saved.

*YES   The save operation ends before any data is written to the media if any configuration objects or system resource manager objects cannot be saved.

Allow object differences (ALWOBJDIF)

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

Notes:

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

The types of differences include:

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

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

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

Single values

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

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

Other values (up to 3 repetitions)

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

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

*OWNER

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

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

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

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

<div align="right">Top</div>

Examples

Example 1: Saving and Restoring Generic Objects

```
SAVRSTCFG   OBJ(ABCD*)  RMTLOCNAME(SYSTEM1)  OBJTYPE(*CFGL)
            ALWOBJDIF(*NONE)
```

This command saves and restores all objects whose names begin with ABCD and whose object type is configuration list (*CFGL). Only those objects that have no differences on SYSTEM1 will be restored.

Example 2: Saving and Restoring All Objects

```
SAVRSTCFG   OBJ(*ALL)  RMTLOCNAME(SYSTEM1)
```

This command saves and restores all configuration objects whose types match the five object types listed on the OBJTYPE parameter.

<div align="right">Top</div>

Error messages

<u>***ESCAPE Messages**</u>

CPCAD80

&1 objects saved and restored.

CPFAD8D

An error occurred during the &1 operation.

CPFAD80

Unable to establish connection from &1 to &2.

CPFAD81

User profile &1 not found on remote location &2.

CPFAD82

Remote location &1 not found.

CPFAD83

Remote location &1 cannot be source location.

CPFAD84

        ObjectConnect internal error, function code &1, return code &2.

CPFAD86

        Location name &1 unable to close &2.

CPFAD88

        Unable to establish connection from &1 to &2.

CPFAD93

        APPC failure. Failure code is &3.

CPF389C

        ObjectConnect internal error, function code &1, return code &2.

Top

Save Restore Changed Objects (SAVRSTCHG)

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

The Save/Restore Changed Object (SAVRSTCHG) command saves and concurrently restores a copy of each changed object, or group of objects located in the same library, to another system. For database files, only the changed members are saved and restored. Objects or members changed since the specified date and time are saved and restored.

Objects changed since the specified date and time are saved and restored with the following exceptions:

- If OBJJRN(*NO) is specified, objects currently being journaled are not saved and restored, unless journaling was started after the specified date and time. This ensures that changes made to objects before journaling starts are not lost (because they were not journaled in a journal receiver).
- Freed objects (programs, files, journal receivers, and so forth) are not saved.
- User-defined messages, job and output queue definitions, logical file definitions, and data queue descriptions are saved and restored, but the contents of those objects are not saved and restored. Logical file access paths are saved and restored if ACCPTH(*YES) is specified.

Specified objects that were changed, and the libraries where they reside, remain locked during the save and restore operation.

To determine the date and time that an object was changed, run the Display Object Description (DSPOBJD) command with DETAIL(*FULL) specified. For database file members that were changed, run the Display File Description (DSPFD) command.

The types of objects that can be saved and restored by this command are listed in the OBJTYPE parameter description in ″Commonly used parameters: Expanded descriptions″ in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter. The system saves and concurrently restores the changed objects.

Restrictions:

1. To use this command, the user must have the special authority *SAVSYS specified in the user profile by the SPCAUT parameter. Otherwise, the user must have object existence authority for each object specified, and execute authority to the specified library. If the user does not have the necessary authority to a specified object, all changed objects except that object are saved and restored.
2. No changed object that is being saved and restored can be changed by a job that is running when the save and restore operation occurs unless save-while-active is used.
3. When the contents of a save file are saved with SAVFDTA(*YES), the save file must be restored before objects contained in it can be restored.
4. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 1
LIB	Saved library	Single values: *ALLUSR Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 2
RMTLOCNAME	Remote location name	*Name*	Required, Positional 3
OBJTYPE	Object types	Single values: ***ALL** Other values (up to 72 repetitions): *ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *JOBD, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *ORTBL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRTIMG, *PSFCFG, *QMFORM, *QMQRY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *USRIDX, *TIMZON, *USRQ, *USRSPC, *VLDL, *WSCST	Optional
OBJJRN	Journaled objects	***NO**, *YES	Optional
REFDATE	Reference date	*Date*, ***SAVLIB**	Optional
REFTIME	Reference time	*Time*, ***NONE**	Optional
STRLIB	Starting library	*Name*, ***FIRST**	Optional
OMITLIB	Libraries to omit	Single values: ***NONE** Other values (up to 300 repetitions): *Generic name, name*	Optional
OMITOBJ	Objects to omit	Values (up to 300 repetitions): *Element list*	Optional
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Generic name, name*, *ALL, ***NONE**	
	Qualifier 2: Library	*Generic name, name*, ***ALL**	
	Element 2: Object type	*Character value*, ***ALL**, *ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *JOBD, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *M36, *M36CFG, *NODGRP, *NODL, *ORTBL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PSFCFG, *QMFORM, *QMQRY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TIMZON, *TBL, *USRIDX, *USRQ, *USRSPC, *VLDL, *WSCST	
ASPDEV	ASP device	*Name*, *****, *SYSBAS, *CURASPGRP	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
PRECHK	Object pre-check	***NO**, *YES	Optional
SAVACT	Save active	***NO**, *LIB, *SYSDFN	Optional
SAVACTWAIT	Save active wait time	*Element list*	Optional
	Element 1: Object locks	0-99999, **120**, *NOMAX	
	Element 2: Pending record changes	0-99999, ***LOCKWAIT**, *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, ***LOCKWAIT**, *NOMAX	

Keyword	Description	Choices	Notes
SAVACTMSGQ	Save active message queue	*Qualified object name*	Optional
	Qualifier 1: Save active message queue	Name, ***NONE**, *WRKSTN	
	Qualifier 2: Library	Name, ***LIBL**, *CURLIB	
ACCPTH	Save access paths	***SYSVAL**, *NO, *YES	Optional
SAVFDTA	Save file data	***YES**, *NO	Optional
QDTA	Queue data	***NONE**, *DTAQ	Optional
RSTLIB	Restore to library	Name, ***LIB**	Optional
RSTASPDEV	Restore to ASP device	Name, ***SAVASPDEV**	Optional
RSTASP	Restore to ASP number	1-32, ***SAVASP**	Optional
OPTION	Option	***ALL**, *NEW, *OLD	Optional
MBROPT	Data base member option	***MATCH**, *ALL, *NEW, *OLD	Optional
ALWOBJDIF	Allow object differences	Single values: ***NONE**, *ALL Other values (up to 3 repetitions): *AUTL, *OWNER, *PGP	Optional
FRCOBJCVN	Force object conversion	Single values: ***SYSVAL**, *NO Other values: *Element list*	Optional
	Element 1: Convert during restore	***YES**	
	Element 2: Objects to convert	***RQD**, *ALL	

Objects (OBJ)

Specifies the names of one or more objects, or the generic names of each group of objects, to check for changes and then to save those objects that have changed. All the objects must be in the library specified for the **Library (LIB)** parameter. If *ALL is specified or defaulted for the **Object types (OBJTYPE)** parameter, all the object types listed in the description of that parameter are saved, provided they are in the specified library and have the specified names.

This is a required parameter.

Single values

*ALL All changed objects in the specified libraries are saved, depending on the values specified for the OBJTYPE parameter.

Other values (up to 300 repetitions)

generic-name
Specify one or more generic names of groups of changed objects to save in the specified library. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete object name.

name Specify the names of specific objects to save. Both generic names and specific names can be specified in the same command.

Saved library (LIB)

Specifies the library that contains the changed objects to be saved and restored.

Single values

*ALLUSR

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

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

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

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

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

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. Up to 300 generic library values can be specified.

name   Specify the name of the library to be saved and restored. Up to 300 library names can be specified.

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format cccccccc or nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name.

remote-location-name

Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name
>   Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Object types (OBJTYPE)

Specifies the types of system objects whose changes are saved. The object types saved are also the ones saved and restored by the Save Library (SAVLIB), Restore Object (RSTOBJ), and Restore Library (RSTLIB) commands. Data dictionaries and the associated files are saved only by using the SAVLIB command.

Single values

*ALL   Changes to all object types that are specified by name, and which are in the specified library, are saved.

Other values (up to 300 repetitions)

object-type
>   Specify the value for each of the types of objects that are saved, such as command (*CMD), file (*FILE), or program (*PGM).
>
>   To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Journaled objects (OBJJRN)

Specifies whether to save changed objects that are currently being journaled and that have been journaled since the date and time specified for the **Reference date (REFDATE)** and **Reference time (REFTIME)** parameters.

*NO   Objects being journaled are not saved. If journaling was started after the specified date and time, the changed objects or changed database file members are saved. The date and time of the last journal start operation can be shown by using the Display Object Description (DSPOBJD) command.

*YES   Objects whose changes are entered in a journal are saved.

Reference date (REFDATE)

Specifies the reference date. Objects that have been changed since this date are saved.

*SAVLIB
>   The objects that have been changed since the date of the last running of the Save Library (SAVLIB) command are saved. If the specified library was never saved, a message is issued and the library is not saved, but the operation continues.

date   Specify the reference date; objects that have been changed since this date are saved. If you specify a date later than the date of the running of this command, a message is issued and the operation ends. The date must be specified in the job date format.

Reference time (REFTIME)

Specifies the reference time. Objects that have been changed since this time on the specified date are saved.

*NONE

No explicit time is specified. Any objects changed since the date specified for the **Reference date (REFDATE)** parameter are saved.

time   Specify the reference time; objects that have been changed since this time on the specified date are saved. If *SAVLIB is specified for the REFDATE parameter, no reference time can be specified. If you specify a time later than the time of the running of this command, a message is issued and the operation ends.

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

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

Starting library (STRLIB)

Specifies the library with which to begin the *ALLUSR save. This parameter is used to recover from ended or failed *ALLUSR save operations.

*FIRST

The save operation begins with the first library in alphabetical order.

name   Specify the name of the library with which to begin the save.

Libraries to omit (OMITLIB)

Specifies the names of one of more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

*NONE

No libraries are excluded from the save operation.

Other values (up to 300 repetitions)

generic-name

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

name   Specify the name of the library to be excluded from the save operation.

Objects to omit (OMITOBJ)

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

Element 1: Object

Qualifier 1: Object

*NONE
> No objects are excluded from the operation.

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

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

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

Qualifier 2: Library

*ALL   The specified objects are excluded from all libraries that are part of the operation.

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

name   Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

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

character-value
> Specify the object type of the objects to be excluded from the operation.
>
> To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

* The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS
    The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP
    If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

*CURRENT
    The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value
    Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:

1. The objects do not exist
2. The library or the objects were previously found to be damaged
3. The library or the objects are locked by another job
4. The requester of the save operation does not have authority for the library or to save the objects.

*NO The save operation for a library continues, saving only those objects that can be saved.

*YES If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation

continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is performed as if SAVACT(*NO) was specified.

*NO   Objects that are in use are not saved. Objects cannot be updated while being saved.

*LIB   Objects in a library can be saved while they are in use by another job. All the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

*SYSDFN
>   Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an object remains in use for the specified time, the object is not saved.

120   The system waits up to 120 seconds for each individual object lock before continuing the save operation.

*NOMAX
>   No maximum wait time exists.

0-99999
>   Specify the number of seconds to wait for each individual object lock before continuing the save operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)** parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value *NOCMTBDY is specified.

*LOCKWAIT
>   The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

*NOCMTBDY

The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify *NOCMTBDY.

Note: This value cannot be specified if the **Target release (TGTRLS)** parameter value is earlier than V5R3M0.

*NOMAX

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.
- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

*LOCKWAIT

The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

*NOMAX

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.

If 0 is specified, and only one name is specified for the **Objects (OBJ)** parameter, and *FILE is the only value specified for the **Object types (OBJTYPE)** parameter, the system will save the object without requiring the types of transactions that are listed above to reach a commit boundary.

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE
> No notification message is sent.

*WRKSTN
> The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name   Specify the name of the message queue to be used.

Qualifier 2: Library

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

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

name   Specify the name of the library where the message queue is located.

Save access paths (ACCPTH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Backup and Recovery book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

*SYSVAL
> The QSAVACCPTH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

*NO   Only those objects specified on the command are saved. No logical file access paths are saved.

*YES   The specified physical files and all eligible logical file access paths over them are saved.

> **Note:** Specifying this value does not save the logical files.

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

*YES   The description and contents of a save file are saved.

*NO   Only the description of a save file is saved.

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

*NONE
> Only the description of a queue is saved.

*DTAQ
> The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Restore to library (RSTLIB)

Specifies the name of the library to which the objects are restored.

*LIB   The restore library is the same name as the value specified on the LIB parameter.

name   Specify the name of the library to which the objects are restored.

Restore to ASP device (RSTASPDEV)

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

Note: You can specify either the RSTASPDEV parameter or the RSTASP parameter, but not both.

*SAVASPDEV
> The data is restored to the same ASP from which it was saved.

name   Specify the name of the ASP device to be used.

Restore to ASP number (RSTASP)

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

Some objects cannot be restored to user ASPs. More information about object types which can be restored to user ASPs is in the Backup and Recovery book, SC41-5304. If the library exists in, or is being restored to the system ASP, journals, journal receivers, and save files can be restored to basic user ASPs. All other object types will be restored to the ASP of the library.

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

*SAVASP
>   The objects are restored to the ASP from which they were saved.

1-32   Specifies the ASP number. When the specified ASP is 1, the specified objects are restored to the system ASP, and when the specified ASP is 2 through 32, the objects are restored to the basic user ASP specified.

Option (OPTION)

Specifies how to handle restoring each object.

*ALL   All the objects in the saved library are restored to the library. Objects in the saved library replace the current versions in the system library. Objects not having a current version are added to the system library. Objects presently in the library, but not on the media, remain in the library.

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

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

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

Data base member option (MBROPT)

Specifies, for database files that exist on the system, which members are restored. If *MATCH is used, the member list in the saved file must match, member for member, the current version on the system. All members are restored for files that do not exist, if the file is restored.

*MATCH
>   The saved members are restored if the lists of the members where they exist match, member for member, the lists of the current system version. MBROPT(*MATCH) is not valid when *ALL is specified for the **Allow object differences (ALWOBJDIF)** parameter.

*ALL   All members in the saved file are restored.

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

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

Allow object differences (ALWOBJDIF)

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

Notes:

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

The types of differences include:

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

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

- File level id: The creation date and time of the database file on the system does not match the creation date and time of the file that was saved.
- Member level id: The creation date and time of the database file member on the system does not match the creation date and time of the member that was saved.
- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.
- Primary Group: The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

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

*ALL All of the differences listed above are allowed on the restore operation. File level id and member level id differences are handled differently than the *FILELVL value. If there is a file level difference and *ALL is specified for the **Data base member option (MBROPT)** parameter, the existing version of the file is renamed and the saved version of the file is restored. If there is a member level difference, the existing version of the member is renamed and the saved version of the member is restored. This value will restore the saved data, but the result may not be correct. You will need to choose whether the restored data or the renamed data is correct, and you will need to make the necessary corrections to the database. For other differences, see the description of each individual value to determine how differences are handled.

Other values (up to 4 repetitions)

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

exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

*FILELVL

File level id and member level id differences are allowed. An attempt will be made to restore existing physical files even though the physical file on the save media may have a different file level id or member level id than the physical file on the system. The physical file data will only be restored for those physical files whose format level identifiers on the save media match the format level identifiers of the corresponding physical file on the system.

If this value is not specified, file level id and member level id differences are not allowed. If an object already exists on the system with a different file level id or member level id than the saved object, the object is not restored.

*OWNER

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

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

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

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

Force object conversion (FRCOBJCVN)

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

Notes:

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

Single values

*SYSVAL

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

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

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

Element 1: Convert during restore

*YES The objects are converted during the restore operation.

Notes:

1. If FRCOBJCVN(*YES *RQD) is specified, then the QFRCCVNRST system value must have a value of ″0″, ″1″, or ″2″. FRCOBJCVN(*YES *RQD) will override a QFRCCVNRST value of ″0″

or "1". If FRCOBJCVN(*YES *ALL) is specfied, then QFRCCVNRST can have any valid value
and FRCOBJCVN(*YES *ALL) overrides the QFRCCVNRST system value.

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

Element 2: Objects to convert

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

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

Examples

Example 1: Saving and Restoring Multiple Changed Objects

```
SAVRSTCHG   OBJ(*ALL)  RMTLOCNAME(SYSTEM1)  LIB(*ALLUSR)
            OBJTYPE(*PGM)  STRLIB(MORE)  ALWOBJDIF(*NONE)
```

This command saves all changed objects whose type is *PGM and are located in user libraries beginning
with the library named MORE. The objects are restored on the remote system named SYSTEM1.

Example 2: Saving and Restoring a Specific Changed Object

```
SAVRSTCHG   OBJ(PETE)  RMTLOCNAME(SYSTEM1)  LIB(RON)
            ALWOBJDIF(*ALL)  OPTION(*NEW)
```

This command saves changed objects named PETE of all types supported, which is located in the library
named RON, and restores the object on the remote system named SYSTEM1 if the object does not already
exist on the remote system.

Error messages

*ESCAPE Messages

CPCAD80

&1 objects saved and restored.

CPFAD8D

An error occurred during the &1 operation.

CPFAD80

Unable to establish connection from &1 to &2.

CPFAD81

User profile &1 not found on remote location &2.

CPFAD82

Remote location &1 not found.

CPFAD83

Remote location &1 cannot be source location.

CPFAD84

    ObjectConnect internal error, function code &1, return code &2.

CPFAD86

    Location name &1 unable to close &2.

CPFAD88

    Unable to establish connection from &1 to &2.

CPFAD93

    APPC failure. Failure code is &3.

CPF389C

    ObjectConnect internal error, function code &1, return code &2.

Top

Save Restore Doc/Lib Object (SAVRSTDLO)

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

The Save/Restore Document Library Object (SAVRSTDLO) command saves and restores the following objects to another system. The system must have a supported communication link with the system that the objects are being restored to.

- Documents
- Folders
- Distribution objects (mail)

Notes:

1. When a folder is saved and restored, the folder object is transferred along with the documents contained in that folder and the subfolders and documents in the subfolders and all successively nested folders and documents. Specific folders can be saved and restored individually using the DLO(*FLRLVL) parameter.

2. Distribution objects (mail) cannot be sent for individual users. Mail can be saved and restored for all users only.

3. SAVRSTDLO does not require a dedicated system; however, individual objects in use when the save and restore command is issued cannot be saved and restored. To ensure all document library objects are saved and restored, run this command when no office activity is occurring on the system.

Restrictions:

1. You must have *ALLOBJ or *SAVSYS special authority to use the following parameter combinations on this command:
   - DLO(*ALL) FLR(*ANY)
   - DLO(*CHG)
   - DLO(*MAIL)
   - DLO(*SEARCH) OWNER(*ALL)
   - DLO(*SEARCH) OWNER(*user-profile-name*)

   where the *user profile name* specified is not the *user profile name* of the user issuing the SAVRSTDLO command.

2. Users that do not have *ALLOBJ or *SAVSYS special authority must:
   - Have *ALL authority for each document or folder to be sent.
   - Be enrolled as Document Interchange Architecture (DIA) users.

3. This command cannot be used while another job is running commands such as RCLDLO, SAVDLO, and RSTDLO because exclusive use of internal objects may have been obtained by these commands.

4. Determining document or folder ownership does not include checking group profiles if one is associated with the specified user profile.

5. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Single values: *ALL, *SEARCH, *CHG, *DOCL, *MAIL, *FLRLVL Other values (up to 300 repetitions): *Character value*	Required, Positional 1
RMTLOCNAME	Remote location name	*Name*	Required, Positional 2
FLR	Folder	Single values: ***ANY**, *NONE Other values (up to 300 repetitions): *Character value*	Optional
OMITFLR	Folders to omit	Single values: ***NONE** Other values (up to 300 repetitions): *Character value*	Optional
SRCHTYPE	Search type	***DOC**, *ALL	Optional
CHKEXP	Expiration date	*Date*, ***NO**, *CURRENT	Optional
CRTDATE	Creation date	*Element list*	Optional
	Element 1: Starting time and date	*Element list*	
	Element 1: Starting time	*Time*, ***AVAIL**	
	Element 2: Starting date	*Date*, ***BEGIN**, *CURRENT	
	Element 2: Ending time and date	*Element list*	
	Element 1: Ending time	*Time*, ***AVAIL**	
	Element 2: Ending date	*Date*, ***END**	
DOCCLS	Document class	*Character value*, ***ANY**	Optional
OWNER	Owner profile	*Name*, ***CURRENT**, *ALL	Optional
REFCHGDATE	Last changed date	*Date*, ***ANY**, *SAVDLOALL	Optional
REFCHGTIME	Last changed time	*Time*, ***ANY**	Optional
DOCL	Document list	*Character value*, ***NONE**	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
STG	Storage	***KEEP**, *DELETE, *FREE	Optional
CMDCHRID	Command character identifier	Single values: ***SYSVAL**, *DEVD Other values: *Element list*	Optional
	Element 1: Graphic character set	*Integer*	
	Element 2: Code page	*Integer*	
SAVACT	Save active	***NO**, *YES	Optional
SAVACTWAIT	Save active wait time	0-99999, **120**, *NOMAX	Optional
ASP	Auxiliary storage pool ID	1-32, ***ANY**	Optional
NEWOBJ	Object name generation	***SAME**, *NEW	Optional
ALWOBJDIF	Allow object differences	***NONE**, *ALL	Optional
RSTASP	Restore to ASP number	1-32, ***SAVASP**	Optional

Top

Document library object (DLO)

Specifies the document library objects to save. To save a folder, DLO(*ALL) must be specified.

This is a required parameter.

Single values

*ALL All document library objects further qualified by the FLR parameter are to be saved. Specifying DLO(*ALL) FLR(*ANY) saves all document library objects.

*SEARCH
> All document library objects that meet the specified search values are saved. Search values are specified by using the following parameters:
> - **Folder (FLR)** parameter.
> - **Check for mark (CHKFORMRK)** parameter.
> - **Expiration date (CHKEXP)** parameter.
> - **Creation date (CRTDATE)** parameter.
> - **Document class (DOCCLS)** parameter.
> - **Owner profile (OWNER)** parameter.
> - **Last changed date (REFCHGDATE)** parameter.
> - **Last changed time (REFCHGTIME)** parameter.
>
> **Note:** Folders are saved only if SRCHTYPE(*ALL) is specified.

*CHG All documents created or changed and all folders created since the last complete save operation and all mail is saved.

*SYSOBJNAM
> The documents with the system object names specified for the **System object name (SYSOBJNAM)** parameter are saved.

*DOCL
> The list of documents referred to in a document list specified for the **Document list (DOCL)** parameter is saved.

*MAIL
> The distribution objects and documents referred to by a mail log are saved.

*FLRLVL
> The folders specified for the **Folder (FLR)** parameter and documents in the folders are saved. Subfolders are not saved.

Other values (up to 300 repetitions)

document-name
> Specify the user-assigned names of the documents that are saved. All documents specified must be in the same folder and that folder must be specified for the **Folder (FLR)** parameter.

Note: System object name is not supported.

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format cccccccc or nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name.

remote-location-name
> Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name
> Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

<div align="right">Top</div>

Folder (FLR)

Specifies the name of the folder to save.

Single values

*ANY Document library objects can be saved from any folder. Consider the following when using the FLR parameter:
- FLR(*ANY) is not valid when one of the following is specified:
  - DLO(*DOCL)
  - DLO(*FLRLVL)
  - DLO(*document-name*)
- FLR(*ANY) is required when one of the following is specified:
  - DLO(*CHG)
  - DLO(*SYSOBJNAM)
  - DLO(*MAIL)
  - DLO(*SEARCH) SRCHTYPE(*ALL)
- When SAVDLO DLO(*ALL) FLR(*ANY) is specified, the following are saved:
  - All documents
  - All folders
  - All distribution objects (mail)

*NONE
> The documents saved are not in any folder. FLR(*NONE) is valid only when one of the following is specified:
- DLO(*ALL)
- DLO(*SEARCH) SRCHTYPE(*DOC)

Other values (up to 300 repetitions)

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

folder-name
> Specify the user-assigned name of the folder in which the documents to be saved are located. The folder name can be a maximum of 63 characters in length.
- Folder objects specified here are saved only when DLO(*ALL) or DLO(*FLRLVL) is specified.
- FLR(*folder-name*) is not valid when one of the following is specified:

– DLO(*SYSOBJNAM)
– DLO(*MAIL)
– DLO(*SEARCH) SRCHTYPE(*ALL)
- Only one folder name can be specified when one of the following is specified:
  – DLO(*DOCL)
  – DLO(*SEARCH) SRCHTYPE(*DOC)
  – DLO(*document-name*)

Note: System object name is not supported.

Folders to omit (OMITFLR)

Specifies the names of one of more folders, or the generic names of each group of folders, to be excluded from the save operation.

Single values

*NONE
No folders are excluded from the save operation.

Other values (up to 300 repetitions)

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

folder-name
Specify the name of the folder to be excluded from the save operation.

Note: System object name is not supported.

Search type (SRCHTYPE)

Specifies the type of objects for which to search. This parameter is valid only if *SEARCH is specified for the **Document library object (DLO)** parameter.

*DOC Only documents are to be searched and saved.

*ALL Documents and folders are to be searched and saved.

Expiration date (CHKEXP)

Specifies that all documents with an expiration date before the specified date are to be saved. The expiration date is assigned by the user when a document is created to specify when the document is no longer needed. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter and *DOC is specified for the **Search type (SRCHTYPE)** parameter.

*NO   The expiration date is ignored.

*CURRENT

> All documents with an expiration date before today's date are saved.

date   Specify a document expiration date. All documents with an expiration date before this date are saved.

Creation date (CRTDATE)

Specifies that documents and folders that have a creation date during the time period specified are to be saved. The time period is specified by a starting time and date and an ending time and date. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter. The time period is specified as follows:

```
((start-time  start-date) (end-time  end-date))
```

Element 1: Starting time and date

Element 1: Starting time

Use one of the following to specify the starting time. Documents must have been created after this time to be selected. Documents created before this time are not selected.

*AVAIL

> Documents and folders filed at any time are eligible for saving.

time   Specify the starting time. When the starting time is used as a search value, the starting date must not be *BEGIN. The starting-time must be the same as the value specified for the **Last changed time (REFCHGTIME)** parameter when the REFCHGTIME parameter is specified. The time can be specified with or without a time separator:

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

Element 2: Starting date

Use one of the following to specify the starting date. Documents must have been created on or after this date to be saved. Documents created before this date are not saved.

*BEGIN

> Documents and folders are saved regardless of the creation date associated with the object.

*CURRENT

> Only documents and folders filed on today's date, after the starting time (if specified), are selected.

date   Specify a starting date for the document creation date time period. The date must be

entered in the job date format. The start date must be the same as the value specified for the **Last changed date (REFCHGDATE)** parameter when the REFCHGDATE parameter is specified.

Element 2: Ending time and date

Element 1: Ending time

Use one of the following to specify the ending time. Documents must have been created before this time to be saved. Any documents created after the specified time are not saved.

*AVAIL
> Documents and folders filed at any time are selected for saving.

time   Specify the ending time. When the ending time is to be used as a search value, the ending date must not be *END. See the description of *starting-time* for details about how time can be specified.

Element 2: Ending date

Use one of the following to specify the ending date. Documents must have been created on or before this date to be saved. Documents created after this date are not saved.

*END   Documents and folders filed on any date are selected. The ending time is not allowed when *END is specified.

date   Specify the ending date for the document creation date time period. Documents created on or before this date are saved. The date must be specified in job date format.

Document class (DOCCLS)

Specifies the class of documents to be saved. The class is assigned by the user when the document is created. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter and *DOC is specified for the **Search type (SRCHTYPE)** parameter.

Note: Although document classes are user-assigned, double-byte character set (DBCS) data cannot be specified on this parameter.

*ANY   The document class is not used to select documents for saving.

character-value
> Specify the document class, ranging from 1 through 16 characters, used to select documents for saving.

Owner profile (OWNER)

Specifies the owner of the documents and folders to be saved. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter.

*CURRENT
> Documents and folders owned by the current requester are saved.

*ALL   This parameter is not used to select documents and folders for saving. You must have all object (*ALLOBJ) or save system (*SAVSYS) special authority if *ALL is specified.

name   Specify the name of the user profile that owns the documents and folders to be saved. All documents and folders owned by this user and that meet the other search values specified are saved. *ALLOBJ or *SAVSYS special authority is required if a user profile is selected other than the user profile of the user issuing this command.

Last changed date (REFCHGDATE)

Specifies the date after which the folders that are created and the documents that are changed or created are to be saved. The change date is updated when the document content or description is changed. This parameter is valid only when *SEARCH is specified for the **Document library object(DLO)** parameter.

*ANY   No reference change date is specified. Documents are saved regardless of the date they were created or changed. Folders are saved regardless of the date they were created.

*SAVDLOALL
> Folders that have been created and documents created or changed since that last complete save operation are saved.

date   Specify the date after which the created folders or the created or changed documents are saved.

Last changed time (REFCHGTIME)

Specifies the time, relative to the date specified for the REFCHGDATE parameter, after which the folders that are created and the documents that are changed or created are to be saved. The change time is updated when the document content or description is changed. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter.

*ANY   No time is specified. The documents are saved regardless of the time they were created or changed. Folders are saved regardless of the time they were created.

time   Specify the time after which the created folders or the created or changed documents are saved.

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

> - Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.

> - With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Document list (DOCL)

Specifies a list of documents to be saved. The document list must be in a folder. The name of the folder must be specified using the **Folder (FLR)** parameter. You must have use (*USE) authority to the folders containing the documents in the document list.

Note: A document list must be the result of a local search, not a remote search.

*NONE
> No document list is saved.

character-value
> Specify the document list to be saved.

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

*CURRENT
> The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value
> Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Storage (STG)

Specifies whether system storage occupied by the document being saved is kept, deleted, or freed after the save operation ends.

Note: STG(*DELETE) and STG(*FREE) are not valid when any of the following are specified:
- DLO(*ALL) FLR(*ANY)
- DLO(*SEARCH) CHKFORMRK(*YES)
- DLO(*CHG)
- DLO(*MAIL)
- SAVACT(*YES)

*KEEP
> The storage occupied by the document remains unchanged after the save operation.

*DELETE
> The document object and all search terms are deleted from the system after the save operation.

*FREE The document description and search terms remain on the system but the storage occupied by the document is deleted after the save operation. The document cannot be used until the document is restored to the system.

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data specified for the **Document class (DOCCLS)** parameter. The character identifier is related to the display device used to enter the document class.

Single values

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

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

Element 1: Graphic character set

1-32767
> Specify the graphic character set to use.

Element 2: Code page

1-32767
> Specify the code page to use.

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

*NO Document library objects in use are not saved. Document library objects cannot be updated while being used.

*YES Document library objects can be changed during the save request.

> **Note:** Some applications update document library objects directly. The data is supplied to the application rather than saving the updates in a temporary file and then updating the DLOs. DLOs that are being updated directly (typically, those being updated by PC-based applications) will not be saved. See the Backup and Recovery information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for more information on using this parameter.

Save active wait time (SAVACTWAIT)

Specifies the amount of time for an object that is in use, before continuing the save operation. If an object remains in use for the specified time, the object is not saved.

120 The system waits up to 120 seconds for each individual object to become available before continuing the save operation.

*NOMAX
    No maximum wait time exists.

0-99999
    Specify the number of seconds to wait for each individual object before continuing the save operation.

<div align="right">Top</div>

ASP number (ASP)

Specifies the number of the auxiliary storage pool (ASP) of the document library object (DLO) to be saved.

*ANY The objects to be saved reside in any ASP. When DLO(*ALL) FLR(*ANY) are specified, all document library objects on the system are saved.

    Note: When DLOs from multiple ASPs are saved, multiple tape media files are created. The beginning and ending sequence numbers of these media files will be required on the RSTDLO command to restore all ASPs.

1-32 Specify the number of an existing ASP that contains the document library objects to be saved. ASP 1 is the system ASP.

    Note: Mail that has not been filed and documents that are not in a folder reside in the system ASP.

<div align="right">Top</div>

Object name generation (NEWOBJ)

Specifies whether a new library-assigned name and system object name are generated for the folders and documents that are restored.

*SAME
    The library-assigned name and system object name do not change.

*NEW A new library-assigned name and system object name are generated for each document or folder that is restored.

<div align="right">Top</div>

Allow object differences (ALWOBJDIF)

Specifies whether the following differences encountered during a restore operation are allowed.

- Ownership—the owner of the object on the system is different than the owner of the object from the save operation.
- System object name—the system object name on the system does not match the system object name on the media.

- Authorization list linking—the object is being restored to a system different from the one on which it was saved.

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

Note: To use this parameter, you need all object (*ALLOBJ) authority.

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

*ALL All of the differences described above are allowed for the restore operation. An informational message is sent, and the object is restored.

> **Notes:**
> - If the owners of the object do not match, the object is restored, but it will keep the ownership and authorities of the object on the system before the restore operation.
> - If *ALL is specified on this parameter, *NEW cannot be specified for the **Object name generation (NEWOBJ)** parameter.
> - If you are restoring objects to a system different from the one on which they were saved and the objects are secured by an authorization list, specifying *ALL automatically links the objects to the authorization list. If the authorization list does not exist on the new system, a message that includes the name of the missing list is issued.

<div align="right">Top</div>

Restore to ASP number (RSTASP)

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

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

1-32 Specify the number of the ASP in which restored documents and folders are placed.

<div align="right">Top</div>

Examples

Example 1: Saving and Restoring All Document Library Objects

```
SAVRSTDLO   DLO(*ALL)  RMTLOCNAME(SYSTEM1)  FLR(*ANY)
            SAVACT(*YES)  NEWOBJ(*NEW)
```

This command saves all document library objects located in any folder and restores the objects to a remote system named SYSTEM1. The objects can be changed during the save and restore operation and new names will be given to the libraries in which the objects are restored.

Example 2: Saving and Restoring Specific Document Library Objects

```
SAVRSTDLO   DLO(*SEARCH)  RMTLOCNAME(SYSTEM2)  FLR(FLR2)
            OWNER(USER8)  SRCHTYPE(*DOC)
```

This command searches for a folder named FLR2 and checks to see if the documents owned by user USER8. The folder and the documents in the folder are saved and restored on a remote system named SYSTEM2.

Error messages

*ESCAPE Messages

CPCAD82

&1 document library objects saved and restored.

CPFAD8C

An error occurred during the SAVRSTDLO operation.

CPFAD80

Unable to establish connection from &1 to &2.

CPFAD81

User profile &1 not found on remote location &2.

CPFAD82

Remote location &1 not found.

CPFAD83

Remote location &1 cannot be source location.

CPFAD84

ObjectConnect internal error, function code &1, return code &2.

CPFAD86

Location name &1 unable to close &2.

CPFAD88

Unable to establish connection from &1 to &2.

CPFAD93

APPC failure. Failure code is &3.

CPF389C

ObjectConnect internal error, function code &1, return code &2.

Save Restore Library (SAVRSTLIB)

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

The Save/Restore Library (SAVRSTLIB) command allows the user to save and restore a copy of one or more libraries to another system. The system must have a supported communication link with the restoring system.

Documents and folders contained in the QDOC library can be saved and restored by using the Save/Restore Document Library Object (SAVRSTDLO) command.

The SAVRSTLIB command saves and restores the entire library; this includes the library description, the object descriptions, and the contents of the objects in the library. For job queues, message queues, and logical files, only the object definitions are saved and restored, not the contents. Logical file access paths can be saved and restored by specifying ACCPTH(*YES). The contents of a save file can be saved and restored by specifying SAVFDTA(*YES). The contents of spooled files on output queues can be saved and restored by specifying SPLFDTA(*ALL). The contents of a data queue can be saved and restored by specifying QDTA(*DTAQ).

The libraries and their objects are not affected on the system unless the command specifies that the storage is to be freed.

The types of objects saved and restored by this command are the same as those listed in the OBJTYPE parameter description in "Commonly used parameters: Expanded descriptions" in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter, with the addition of *DTADCT. Certain I5/OS system objects that are not contained in user libraries (such as user profiles) are not saved and restored by this command. They can be saved by the Save System (SAVSYS) or Save Security Data (SAVSECDTA) commands, and restored by using the Restore User Profile (RSTUSRPRF) command.

Restrictions:

1. To use this command, the user must have either the special authority *SAVSYS specified in the user profile by the SPCAUT parameter, or the user must have:
   - Read authority for, or be the owner of, each library specified.
   - Object existence authority for each object in the library.

   If the user does not have the correct authorities for all of the libraries and objects specified, only those for which the user does have authority are saved and restored.

2. No library that is being saved and restored, or the objects in the library being saved and restored, can be updated by a job that is running at the time the save and restore operation occurs unless save-while-active (SAVACT) is used.

3. When the contents of a save file are saved and restored by specifying SAVFDTA(*YES), the save file must be restored before objects contained in it can be restored.

4. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Single values: *NONSYS, *ALLUSR, *IBM Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 1
RMTLOCNAME	Remote location name	*Name*	Required, Positional 2
STRLIB	Starting library	*Name*, ***FIRST**	Optional
OMITLIB	Libraries to omit	Single values: ***NONE** Other values (up to 300 repetitions): *Generic name, name*	Optional
OMITOBJ	Objects to omit	Values (up to 300 repetitions): *Element list*	Optional
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Generic name, name*, *ALL, ***NONE**	
	Qualifier 2: Library	*Generic name, name*, ***ALL**	
	Element 2: Object type	*Character value*, ***ALL**, *ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *JOBD, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *ORTBL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PSFCFG, *QMFORM, *QMQRY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *USRIDX, *USRQ, *USRSPC, *VLDL, *WSCST	
ASPDEV	ASP device	*Name*, *****, *SYSBAS, *CURASPGRP	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
PRECHK	Object pre-check	***NO**, *YES	Optional
SAVACT	Save active	***NO**, *LIB, *SYSDFN	Optional
SAVACTWAIT	Save active wait time	*Element list*	Optional
	Element 1: Object locks	0-99999, **120**, *NOMAX	
	Element 2: Pending record changes	0-99999, ***LOCKWAIT**, *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, ***LOCKWAIT**, *NOMAX	
SAVACTMSGQ	Save active message queue	*Qualified object name*	Optional
	Qualifier 1: Save active message queue	*Name*, ***NONE**, *WRKSTN	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
ACCPTH	Save access paths	***SYSVAL**, *NO, *YES	Optional
SAVFDTA	Save file data	***YES**, *NO	Optional
SPLFDTA	Spooled file data	***NONE**, *ALL	Optional
QDTA	Queue data	***NONE**, *DTAQ	Optional
STG	Storage	***KEEP**, *FREE	Optional
OPTION	Option	***ALL**, *NEW, *OLD, *FREE	Optional
MBROPT	Data base member option	***MATCH**, *ALL, *NEW, *OLD	Optional
ALWOBJDIF	Allow object differences	Single values: ***NONE**, *ALL Other values (up to 4 repetitions): *AUTL, *FILELVL, *OWNER, *PGP	Optional
RSTLIB	Restore to library	*Name*, ***LIB**	Optional
RSTASPDEV	Restore to ASP device	*Name*, ***SAVASPDEV**	Optional

Keyword	Description	Choices	Notes
RSTASP	Restore to ASP number	1-32, ***SAVASP**	Optional
FRCOBJCVN	Force object conversion	Single values: ***SYSVAL**, *NO Other values: *Element list*	Optional
	Element 1: Convert during restore	***YES**	
	Element 2: Objects to convert	***RQD**, *ALL	

Library (LIB)

Specifies which libraries to save and restore.

Notes:

1. If using *NONSYS, or *IBM keywords some libraries will not be saved and restored, because they are being used for this operation.

2. If the user specifies *ALLUSR on this parameter, this command should be run when the specified libraries are not being used. If objects in a library are in use while the library is being saved and restored, the objects are not saved and restored. To ensure a complete save and restore of all libraries, run the SAVLIB command with the system in a restricted state. For example, if SAVRSTLIB LIB(*ALLUSR) is run when the subsystem QSNADS is active, the QAO* files are not saved and restored in library QUSRSYS. To save and restore the *QAO files, end the QSNADS subsystem before running SAVRSTLIB LIB(*ALLUSR). Some subsystems cannot be ended when using the SAVRSTLIB command. The libraries associated with these subsystems will not be saved and restored using the SAVRSTLIB command.

QSOC if using the optical bus transport with the SAVRSTLIB command

QCMN
> if using the communications transport with the SAVRSTLIB command

*NONSYS
> All user-created libraries, the QGPL and QUSRSYS libraries, and licensed program libraries such as QRPG and QIDU are saved. All subsystems must be ended by the End Subsystem (ENDSBS) or End System (ENDSYS) command before this option is specified. When *NONSYS is specified, the libraries are saved in alphabetical order on the media.

*ALLUSR
> All user libraries are saved and restored. All libraries with names that do not begin with the letter Q are saved and restored except for the following:

> | | | |
> |---|---|---|
> | #CGULIB | #DSULIB | #SEULIB |
> | #COBLIB | #RPGLIB | |
> | #DFULIB | #SDALIB | |

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

> | | | | |
> |---|---|---|---|
> | QDSNX | QRCLxxxxx | QUSRIJS | QUSRVxRxMx |
> | QGPL | QSRVAGT | QUSRINFSKR | |
> | QGPL38 | QSYS2 | QUSRNOTES | |
> | QMGTC | QSYS2xxxxx | QUSROND | |
> | QMGTC2 | QS36F | QUSRPOSGS | |
> | QMPGDATA | QUSER38 | QUSRPOSSA | |

```
QMQMDATA     QUSRADSM     QUSRPYMSVR
QMQMPROC     QUSRBRM      QUSRRDARS
QPFRDATA     QUSRDIRCL    QUSRSYS
QRCL         QUSRDIRDB    QUSRVI
```

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

*IBM   Saves all system (IBM) libraries except for the following Q libraries:

```
QDOC         QRCL         QSYS         QUSRSYS
QDSNX        QRECOVERY    QS36F        QUSRVxRxMx
QGPL         QRPLOBJ      QTEMP
QGPL38       QSPLxxxx     QUSER38
QPFRDATA     QSRV         QUSRINFSKR
```

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

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

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

generic-name
Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. A maximum of 300 generic library names can be specified.

name   Specify the names of the library to be saved and restored. A maximum of 300 library names can be specified.

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format cccccccc or nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name.

remote-location-name
Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name
Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Starting library (STRLIB)

Specifies the library with which to begin the *NONSYS, *IBM, or *ALLUSR save operation.

Note: This parameter is valid only if *NONSYS, *IBM, or *ALLUSR is specified on the **Library (LIB)** parameter.

*FIRST
> The save operation begins with the first library in alphabetical order.

name   Specify the name of the library with which to begin the save operation.

Libraries to omit (OMITLIB)

Specifies the names of one of more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

*NONE
> No libraries are excluded from the save operation.

Other values (up to 300 repetitions)

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

name   Specify the name of the library to be excluded from the save operation.

Objects to omit (OMITOBJ)

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

Element 1: Object

Qualifier 1: Object

*NONE
> No objects are excluded from the operation.

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

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

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

Qualifier 2: Library

*ALL   The specified objects are excluded from all libraries that are part of the operation.

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

name   Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

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

character-value
> Specify the object type of the objects to be excluded from the operation.
>
> To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

<div align="right">Top</div>

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

*   The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS
> The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP
> If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name   Specify the name of the ASP device to be included in the save operation.

<div align="right">Top</div>

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

*CURRENT
> The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value
> Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:

1. The objects do not exist
2. The library or the objects were previously found to be damaged
3. The library or the objects are locked by another job
4. The requester of the save operation does not have authority for the library or to save the objects.

*NO The save operation for a library continues, saving only those objects that can be saved.

*YES If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is performed as if SAVACT(*NO) was specified.

*NO Objects that are in use are not saved. Objects cannot be updated while being saved.

*LIB Objects in a library can be saved while they are in use by another job. All the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

*SYSDFN
> Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an object remains in use for the specified time, the object is not saved.

<u>120</u>
  The system waits up to 120 seconds for each individual object lock before continuing the save operation.

*NOMAX
  No maximum wait time exists.

0-99999
  Specify the number of seconds to wait for each individual object lock before continuing the save operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)** parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value *NOCMTBDY is specified.

<u>***LOCKWAIT**</u>
  The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

*NOCMTBDY
  The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

  If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify *NOCMTBDY.

  Note: This value cannot be specified if the **Target release (TGTRLS)** parameter value is earlier than V5R3M0.

*NOMAX
  No maximum wait time exists.

0-99999
  Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

• Data Definition Language (DDL) object level changes for that library.

- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

*LOCKWAIT
> The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

*NOMAX
> No maximum wait time exists.

0-99999
> Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE
> No notification message is sent.

*WRKSTN
> The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name   Specify the name of the message queue to be used.

Qualifier 2: Library

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

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

name   Specify the name of the library where the message queue is located.

Save access paths (ACCPTH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Backup and Recovery book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

*SYSVAL
> The QSAVACCPTH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

*NO   Only those objects specified on the command are saved. No logical file access paths are saved.

*YES   The specified physical files and all eligible logical file access paths over them are saved.

> **Note:** Specifying this value does not save the logical files.

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

*YES   The description and contents of a save file are saved.

*NO   Only the description of a save file is saved.

Spooled file data (SPLFDTA)

Specifies whether to save spooled file data and attributes for output queues that are saved.

*NONE
> No spooled file data is saved.

*ALL   For each output queue that is saved, all available spooled file data on the output queue is saved.

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

*NONE
Only the description of a queue is saved.

*DTAQ
The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Storage (STG)

Specifies whether the system storage that is occupied by the data portion of the specified members (except for save files), modules, programs, service programs, Structured Query Language (SQL) packages, and journal receivers in the library being saved is freed as part of the save operation. Only the data portion of the objects is freed, not the descriptions of the objects.

*KEEP
The storage occupied by the data portion of the objects being saved is not freed.

*FREE The storage occupied by the data portion of the specified objects being saved is freed as part of the save operation. The storage for all the objects in a library is freed only after all the objects in that library are saved successfully.

Note: To prevent the possible abnormal end of a program, the program being saved must not be running in the system when *FREE is specified.

Option (OPTION)

Specifies how to handle restoring each object.

*ALL All the objects in the saved library are restored to the library. Objects in the saved library replace the current versions in the system library. Objects not having a current version are added to the system library. Objects presently in the library, but not on the media, remain in the library.

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

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

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

Data base member option (MBROPT)

Specifies, for database files that exist on the system, which members are restored. If *MATCH is used, the member list in the saved file must match, member for member, the current version on the system. All members are restored for files that do not exist, if the file is restored.

*MATCH
> The saved members are restored if the lists of the members where they exist match, member for member, the lists of the current system version. MBROPT(*MATCH) is not valid when *ALL is specified for the **Allow object differences (ALWOBJDIF)** parameter.

*ALL   All members in the saved file are restored.

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

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

Allow object differences (ALWOBJDIF)

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

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

The types of differences include:

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

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

- File level id: The creation date and time of the database file on the system does not match the creation date and time of the file that was saved.

- Member level id: The creation date and time of the database file member on the system does not match the creation date and time of the member that was saved.

- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.

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

Single values

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

*ALL   All of the differences listed above are allowed on the restore operation. File level id and member level id differences are handled differently than the *FILELVL value. If there is a file level

difference and *ALL is specified for the **Data base member option (MBROPT)** parameter, the existing version of the file is renamed and the saved version of the file is restored. If there is a member level difference, the existing version of the member is renamed and the saved version of the member is restored. This value will restore the saved data, but the result may not be correct. You will need to choose whether the restored data or the renamed data is correct, and you will need to make the necessary corrections to the database. For other differences, see the description of each individual value to determine how differences are handled.

Other values (up to 4 repetitions)

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

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

*FILELVL
> File level id and member level id differences are allowed. An attempt will be made to restore existing physical files even though the physical file on the save media may have a different file level id or member level id than the physical file on the system. The physical file data will only be restored for those physical files whose format level identifiers on the save media match the format level identifiers of the corresponding physical file on the system.

> If this value is not specified, file level id and member level id differences are not allowed. If an object already exists on the system with a different file level id or member level id than the saved object, the object is not restored.

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

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

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

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

Restore to library (RSTLIB)

Specifies whether the library contents are restored to the same library from which they were saved, or to a different library. If a different library is specified, you cannot specify *NONSYS, *ALLUSR, or *IBM for the **Library (LIB)** parameter.

*LIB   The library contents are restored to the same library or libraries from which they were saved.

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

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

Restore to ASP device (RSTASPDEV)

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

Note: You can specify either the RSTASPDEV parameter or the RSTASP parameter, but not both.

*SAVASPDEV
> The data is restored to the same ASP from which it was saved.

name Specify the name of the ASP device to be used.

Restore to ASP number (RSTASP)

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

Some objects cannot be restored to user ASPs. More information about object types which can be restored to user ASPs is in the Backup and Recovery book, SC41-5304. If the library exists in, or is being restored to the system ASP, journals, journal receivers, and save files can be restored to basic user ASPs. All other object types will be restored to the ASP of the library.

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

*SAVASP
> The objects are restored to the ASP from which they were saved.

1-32 Specifies the ASP number. When the specified ASP is 1, the specified objects are restored to the system ASP, and when the specified ASP is 2 through 32, the objects are restored to the basic user ASP specified.

Force object conversion (FRCOBJCVN)

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

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

Single values

*SYSVAL

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

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

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

Element 1: Convert during restore

*YES The objects are converted during the restore operation.

Notes:

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

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

Element 2: Objects to convert

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

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

<div align="right">Top</div>

Examples

Example 1: Saving and Restoring All User Libraries

```
SAVRSTLIB   LIB(*ALLUSR)   RMTLOCNAME(SYSTEM1)
            STRLIB(GFM1)   OPTION(*OLD)
```

This command saves all user libraries beginning with the GFM1 library and restores them on a remote system named SYSTEM1. Only the objects in the library having a saved version are restored.

Example 2: Saving and Restoring Specific Libraries

```
SAVRSTLIB   LIB(GRUNBOK TIMON VASEK)   RMTLOCNAME(SYSTEM1)
```

This command saves the following libraries GRUNBOK, TIMON and VASEK and restores them on a remote system named SYSTEM1.

<div align="right">Top</div>

Error messages

*ESCAPE Messages

CPCAD81

&1 libraries saved and restored.

CPFAD8B

An error occurred during the SAVRSTLIB operation.

CPFAD80

Unable to establish connection from &1 to &2.

CPFAD81

User profile &1 not found on remote location &2.

CPFAD82

Remote location &1 not found.

CPFAD83

Remote location &1 cannot be source location.

CPFAD84

ObjectConnect internal error, function code &1, return code &2.

CPFAD86

Location name &1 unable to close &2.

CPFAD88

Unable to establish connection from &1 to &2.

CPFAD93

APPC failure. Failure code is &3.

CPF389C

ObjectConnect internal error, function code &1, return code &2.

Top

Save Restore Object (SAVRSTOBJ)

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

The Save/Restore Object (SAVRSTOBJ) command saves and restores a single object, or a group of objects located in the same library, to another system. The objects can be restored to the same library or a different library. The types of objects that can be saved and restored are the same set of objects allowed on the Save Object (SAVOBJ) command. Objects on the system from which the objects are being saved are not affected by the SAVRSTOBJ command.

For job queues, message queues, and logical files, only the object descriptions are saved and restored, not the contents. Logical file access paths can be saved and restored by specifying ACCPTH(*YES). The contents of a save file can be saved and restored by specifying SAVFDTA(*YES). The contents of spooled files on output queues can be saved and restored by specifying SPLFDTA(*ALL). The contents of a data queue can be saved and restored by specifying QDTA(*DTAQ).

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

Restrictions:

1. To use this command, you must have either the special authority *SAVSYS specified in the user profile by the SPCAUT parameter or have (a) object existence authority for each object specified and (b) read authority for the specified library. If you do not have the necessary authority to a specified object, all objects except that one are saved and restored.

2. Both systems intended to participate in the save and restore operation must be connected to the same APPN network, or if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 1
LIB	Saved library	Single values: *ALLUSR Other values (up to 300 repetitions): *Generic name, name*	Required, Positional 2
RMTLOCNAME	Remote location name	*Name*	Required, Positional 3
OBJTYPE	Object types	Single values: ***ALL** Other values (up to 69 repetitions): *ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *JOBD, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *ORTBL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRTIMG, *PSFCFG, *QMFORM, *QMQRY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *USRIDX, *TIMZON, *USRQ, *USRSPC, *VLDL, *WSCST	Optional

Keyword	Description	Choices	Notes
STRLIB	Starting library	*Name*, ***FIRST**	Optional
OMITLIB	Libraries to omit	Single values: ***NONE** Other values (up to 300 repetitions): *Generic name, name*	Optional
OMITOBJ	Objects to omit	Values (up to 300 repetitions): *Element list*	Optional
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Generic name, name*, *ALL, ***NONE**	
	Qualifier 2: Library	*Generic name, name*, ***ALL**	
	Element 2: Object type	*Character value*, ***ALL**, *ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *JOBD, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *M36, *M36CFG, *NODGRP, *NODL, *ORTBL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PSFCFG, *QMFORM, *QMQRY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *USRIDX, *TIMZON, *USRQ, *USRSPC, *VLDL, *WSCST	
ASPDEV	ASP device	*Name*, *****, *SYSBAS, *CURASPGRP	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
PRECHK	Object pre-check	***NO**, *YES	Optional
SAVACT	Save active	***NO**, *LIB, *SYSDFN	Optional
SAVACTWAIT	Save active wait time	*Element list*	Optional
	Element 1: Object locks	0-99999, **120**, *NOMAX	
	Element 2: Pending record changes	0-99999, ***LOCKWAIT**, *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, ***LOCKWAIT**, *NOMAX	
SAVACTMSGQ	Save active message queue	*Qualified object name*	Optional
	Qualifier 1: Save active message queue	*Name*, ***NONE**, *WRKSTN	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
FILEMBR	File member	Values (up to 300 repetitions): *Element list*	Optional
	Element 1: File	*Name*, ***ALL**	
	Element 2: Member	Single values: ***ALL**, *NONE Other values (up to 50 repetitions): *Generic name, name*	
ACCPTH	Save access paths	***SYSVAL**, *NO, *YES	Optional
SAVFDTA	Save file data	***YES**, *NO	Optional
SPLFDTA	Spooled file data	***NONE**, *ALL	Optional
QDTA	Queue data	***NONE**, *DTAQ	Optional
STG	Storage	***KEEP**, *FREE	Optional
OPTION	Option	***ALL**, *NEW, *OLD, *FREE	Optional
MBROPT	Data base member option	***MATCH**, *ALL, *NEW, *OLD	Optional
ALWOBJDIF	Allow object differences	Single values: ***NONE**, *ALL Other values (up to 4 repetitions): *AUTL, *FILELVL, *OWNER, *PGP	Optional
RSTLIB	Restore to library	*Name*, ***LIB**	Optional
RSTASPDEV	Restore to ASP device	*Name*, ***SAVASPDEV**	Optional
RSTASP	Restore to ASP number	1-32, ***SAVASP**	Optional

Keyword	Description	Choices	Notes
FRCOBJCVN	Force object conversion	Single values: ***SYSVAL**, *NO Other values: *Element list*	Optional
	Element 1: Convert during restore	***YES**	
	Element 2: Objects to convert	***RQD**, *ALL	

Objects (OBJ)

Specifies the names of one or more objects or the generic name of each group of objects to be saved. All the objects must be in the library specified for the **Library (LIB)** parameter. If *ALL is specified or defaulted for the **Object types (OBJTYPE)** parameter, all the object types listed in the description of that parameter are saved, provided they are in the specified library and have the specified names.

This is a required parameter.

Single values

***ALL**   All the objects in the specified libraries are saved, depending on the values specified for the OBJTYPE parameter.

Other values (up to 300 repetitions)

generic-name
> Specify one or more generic names of groups of objects in the specified library to be saved. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete object name.

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

Saved library (LIB)

Specifies the library containing the objects to be saved and restored.

Single values

***ALLUSR**

> All user libraries are saved and restored. All libraries with names that do not begin with the letter Q are saved and restored except for the following:

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

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

```
QDSNX      QRCLxxxxx   QUSRIJS     QUSRVxRxMx
QGPL       QSRVAGT     QUSRINFSKR
QGPL38     QSYS2       QUSRNOTES
QMGTC      QSYS2xxxxx  QUSROND
```

```
QMGTC2      QS36F       QUSRPOSGS
QMPGDATA    QUSER38     QUSRPOSSA
QMQMDATA    QUSRADSM    QUSRPYMSVR
QMQMPROC    QUSRBRM     QUSRRDARS
QPFRDATA    QUSRDIRCL   QUSRSYS
QRCL        QUSRDIRDB   QUSRVI
```

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

Other values (up to 300 repetitions)

generic-name
> Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. Up to 300 generic library values can be specified.

name   Specify the name of the library to be saved and restored. Up to 300 library names can be specified.

Top

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format ccccccccc or nnnnnnnn.ccccccccc, where nnnnnnnn is the network identifier (ID) and ccccccccc is the remote location name.

remote-location-name
> Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name
> Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Top

Object types (OBJTYPE)

Specifies the types of system objects to be saved.

Single values

*ALL   All object types that are specified by name and are in the specified library are saved. If *ALL is also specified for the **Objects (OBJ)** parameter, all the objects in the library that are of the types that can be saved are saved.

Other values (up to 300 repetitions)

object-type
> Specify the value for each of the types of objects to be saved, such as command (*CMD), file (*FILE), or program (*PGM).

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

Starting library (STRLIB)

Specifies the library with which to begin the save operation.

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

The basic steps for restarting a save operation are:

1. Check the job log to determine the library where the previous save operation failed. Find the last library saved, which is indicated by a successful completion message.
2. Load the next tape and ensure the tape is initialized.
3. Add the following to your original save command:

   ```
   STRLIB(library-name)  OMITLIB(library-name)
   ```

   where the *library-name* for the STRLIB and OMITLIB parameters is the last library successfully saved. This starts the save operation on the library after the last successfully saved library.

   To restore the libraries, you will need to perform a separate restore operation for each save operation that was performed.

*FIRST
   The save operation begins with the first library value specified for the **Library (LIB)** parameter. If the first value is a generic name or special value, the save operation begins with the first library that matches this value.

name   Specify the name of the library with which to begin the save operation.

Libraries to omit (OMITLIB)

Specifies the names of one of more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

*NONE
   No libraries are excluded from the save operation.

Other values (up to 300 repetitions)

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

name   Specify the name of the library to be excluded from the save operation.

Objects to omit (OMITOBJ)

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

Element 1: Object

Qualifier 1: Object

*NONE
> No objects are excluded from the operation.

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

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

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

Qualifier 2: Library

*ALL   The specified objects are excluded from all libraries that are part of the operation.

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

name   Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

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

character-value
> Specify the object type of the objects to be excluded from the operation.
>
> To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

<div align="right">Top</div>

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

* The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS
The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP
If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

*CURRENT
The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value
Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:

1. The objects do not exist
2. The library or the objects were previously found to be damaged
3. The library or the objects are locked by another job
4. The requester of the save operation does not have authority for the library or to save the objects.

*NO The save operation for a library continues, saving only those objects that can be saved.

*YES If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is performed as if SAVACT(*NO) was specified.

*NO   Objects that are in use are not saved. Objects cannot be updated while being saved.

*LIB   Objects in a library can be saved while they are in use by another job. All the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

*SYSDFN
    Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an object remains in use for the specified time, the object is not saved.

120   The system waits up to 120 seconds for each individual object lock before continuing the save operation.

*NOMAX
    No maximum wait time exists.

0-99999
    Specify the number of seconds to wait for each individual object lock before continuing the save operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)** parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value *NOCMTBDY is specified.

*LOCKWAIT
    The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

*NOCMTBDY
    The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

    If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach

commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify *NOCMTBDY.

Note: This value cannot be specified if the **Target release (TGTRLS)** parameter value is earlier than V5R3M0.

*NOMAX
No maximum wait time exists.

0-99999
Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.
- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

*LOCKWAIT
The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

*NOMAX
No maximum wait time exists.

0-99999
Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.

If 0 is specified, and only one name is specified for the **Objects (OBJ)** parameter, and *FILE is the only value specified for the **Object types (OBJTYPE)** parameter, the system will save the object without requiring the types of transactions that are listed above to reach a commit boundary.

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE
No notification message is sent.

*WRKSTN

> The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name   Specify the name of the message queue to be used.

Qualifier 2: Library

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

*CURLIB

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

name   Specify the name of the library where the message queue is located.

File member (FILEMBR)

Specifies the database file members that are saved. This parameter is made up of two parts: the file name and the member name.

Each database file specified here must also be specified for the **Objects (OBJ)** parameter, by its complete name, a generic name, or *ALL. The **Object types (OBJTYPE)** parameter value must be *ALL or include *FILE.

Note: This parameter cannot be specified when STG(*FREE) is specified.

Element 1: File

*ALL   The list of member name values that follows this value applies to all files specified for the OBJ parameter.

name   Specify the name of the database file from which the listed members are to be saved. Up to 50 files can be specified, with a member list for each file.

> **Note:** Generic names are not valid for the database file name, but are allowed for the member name.

> **Note:** Duplicate file names are not allowed.

Element 2: Member

Single values

*ALL   All members are saved from the specified file.

*NONE

> No members are saved from the specified file. Only the file description is saved.

Other values (up to 50 repetitions)

generic-name

> Specify the generic names of the members to be saved from the specified file. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete member name.

Note: If generic member names are specified, the file must contain member names that match the generic names for the file to be saved. For example, if PAY* is specified as a generic member name, and the system is unable to find a member whose name starts with PAY, the file is not saved. If files specified by the FILEMBR parameter are not saved because members with the specified generic name cannot be found, a diagnostic message is sent, the save operation ends, and an escape message is sent specifying the number of files not saved. If at least one of the files processed for the FILEMBR parameter contains a member with the specified generic name, the diagnostic message is not sent, and the number of files not saved is in the final completion message.

name   Specify the names of the members to be saved from the given file.

> **Note:** If specific member names are specified, the specified members must exist in the file for any part of the file to be saved or restored.

Save access paths (ACCPTH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Backup and Recovery book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

*SYSVAL
> The QSAVACCPTH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

*NO   Only those objects specified on the command are saved. No logical file access paths are saved.

*YES   The specified physical files and all eligible logical file access paths over them are saved.

> **Note:** Specifying this value does not save the logical files.

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

*YES   The description and contents of a save file are saved.

*NO   Only the description of a save file is saved.

Spooled file data (SPLFDTA)

Specifies whether to save spooled file data and attributes for output queues that are saved.

*NONE
        No spooled file data is saved.

*ALL   For each output queue that is saved, all available spooled file data on the output queue is saved.

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

*NONE
        Only the description of a queue is saved.

*DTAQ
        The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Storage (STG)

Specifies whether the system storage that is occupied by the data portion of the specified members (except for save files), modules, programs, service programs, Structured Query Language (SQL) packages, and journal receivers in the library being saved is freed as part of the save operation. Only the data portion of the objects is freed, not the descriptions of the objects.

*KEEP
        The storage occupied by the data portion of the objects being saved is not freed.

*FREE   The storage occupied by the data portion of the specified objects being saved is freed as part of the save operation. The storage for all the objects in a library is freed only after all the objects in that library are saved successfully.

        Note: To prevent the possible abnormal end of a program, the program being saved must not be running in the system when *FREE is specified.

Option (OPTION)

Specifies how to handle restoring each object.

*ALL   All the objects in the saved library are restored to the library. Objects in the saved library replace the current versions in the system library. Objects not having a current version are added to the system library. Objects presently in the library, but not on the media, remain in the library.

*NEW   Only the objects in the saved library that do not exist in the current version of the system library are added to the library. Only objects not known to the system library are restored; known objects

are not restored. This option restores objects that were deleted after they were saved or that are new to this library. If any saved objects have a version already in the system library, they are not restored, and an informational message is sent for each one, but the restore operation continues.

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

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

Data base member option (MBROPT)

Specifies, for database files that exist on the system, which members are restored. If *MATCH is used, the member list in the saved file must match, member for member, the current version on the system. All members are restored for files that do not exist, if the file is restored.

*MATCH

The saved members are restored if the lists of the members where they exist match, member for member, the lists of the current system version. MBROPT(*MATCH) is not valid when *ALL is specified for the **Allow object differences (ALWOBJDIF)** parameter.

*ALL All members in the saved file are restored.

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

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

Allow object differences (ALWOBJDIF)

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

Notes:

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

The types of differences include:

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

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

- File level id: The creation date and time of the database file on the system does not match the creation date and time of the file that was saved.

- Member level id: The creation date and time of the database file member on the system does not match the creation date and time of the member that was saved.
- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.
- Primary Group: The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

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

*ALL All of the differences listed above are allowed on the restore operation. File level id and member level id differences are handled differently than the *FILELVL value. If there is a file level difference and *ALL is specified for the **Data base member option (MBROPT)** parameter, the existing version of the file is renamed and the saved version of the file is restored. If there is a member level difference, the existing version of the member is renamed and the saved version of the member is restored. This value will restore the saved data, but the result may not be correct. You will need to choose whether the restored data or the renamed data is correct, and you will need to make the necessary corrections to the database. For other differences, see the description of each individual value to determine how differences are handled.

Other values (up to 4 repetitions)

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

*FILELVL
> File level id and member level id differences are allowed. An attempt will be made to restore existing physical files even though the physical file on the save media may have a different file level id or member level id than the physical file on the system. The physical file data will only be restored for those physical files whose format level identifiers on the save media match the format level identifiers of the corresponding physical file on the system.
>
> If this value is not specified, file level id and member level id differences are not allowed. If an object already exists on the system with a different file level id or member level id than the saved object, the object is not restored.

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

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

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

Restore to library (RSTLIB)

Specifies the library in which the objects are to be restored.

*LIB The name of the library that was specified on the LIB parameter is used.

name Specify the name of the library to which you want to restore the objects.

Restore to ASP device (RSTASPDEV)

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

Note: You can specify either the RSTASPDEV parameter or the RSTASP parameter, but not both.

*SAVASPDEV
The data is restored to the same ASP from which it was saved.

name Specify the name of the ASP device to be used.

Restore to ASP number (RSTASP)

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

Some objects cannot be restored to user ASPs. More information about object types which can be restored to user ASPs is in the Backup and Recovery book, SC41-5304. If the library exists in, or is being restored to the system ASP, journals, journal receivers, and save files can be restored to basic user ASPs. All other object types will be restored to the ASP of the library.

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

*SAVASP
The objects are restored to the ASP from which they were saved.

1-32 Specifies the ASP number. When the specified ASP is 1, the specified objects are restored to the system ASP, and when the specified ASP is 2 through 32, the objects are restored to the basic user ASP specified.

Force object conversion (FRCOBJCVN)

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

Notes:

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

Single values

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

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

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

Element 1: Convert during restore

*YES   The objects are converted during the restore operation.

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

Element 2: Objects to convert

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

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

Examples

Example 1: Saving and Restoring Generic Objects

```
SAVRSTOBJ   OBJ(ABCD*)  LIB(ACE)  RMTLOCNAME(SYSTEM1)
            OBJTYPE(*PGM)  ALWOBJDIF(*NONE)
```

This command saves the objects beginning with the characters ABCD located in the library named ACE and restores them on the remote system named SYSTEM1.

Example 2: Saving and Restoring a Specific Object

```
SAVRSTOBJ   OBJ(FLETCH)  LIB(CHASE)  RMTLOCNAME(SYSTEM1)
            ALWOBJDIF(*ALL)  OPTION(*NEW)
```

This command saves the object named FLETCH located in the library named CHASE and restores it on the remote system named SYSTEM1 if it is does not already exist on the remote system.

Error messages

<u>*ESCAPE Messages</u>

CPCAD80

      &1 objects saved and restored.

CPFAD8D

      An error occurred during the &1 operation.

CPFAD80

      Unable to establish connection from &1 to &2.

CPFAD81

      User profile &1 not found on remote location &2.

CPFAD82

      Remote location &1 not found.

CPFAD83

      Remote location &1 cannot be source location.

CPFAD84

      ObjectConnect internal error, function code &1, return code &2.

CPFAD86

      Location name &1 unable to close &2.

CPFAD88

      Unable to establish connection from &1 to &2.

CPFAD93

      APPC failure. Failure code is &3.

CPF389C

      ObjectConnect internal error, function code &1, return code &2.

Top

Save S/36 File (SAVS36F)

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

Parameters
Examples
Error messages

The Save System/36 File (SAVS36F) command can create:

- a copy of a single database physical file or logical file to diskette, tape magnetic media or another database physical file in the same format as if a System/36 has saved the file.
- a copy of multiple database physical files or logical files to diskette or tape magnetic media in the same format as if a System/36 had saved the files as a save all set. The files can be all files in a library, all files that are not part of a file group, all files in a specific file group, or all files that begin with a specified set of characters.

A **Save All Set** is a group (set) of files that share the same group (set) name and that are saved (copied) to diskette or tape with one operation. The set of files can be restored (copied back from diskette or tape) with a single operation by referring to the set name (see the Restore System/36 Files (RSTS36F) command).

File groups are defined by file names that contain a period. The characters preceding the period identify the file group, and the characters following the period identify the file within the group. As with file names within the System/36 environment, the maximum number of characters is eight, including the period. Files with names that do not contain a period are not part of a file group. The following examples show the names of files within a file group.

```
PAYROL.A
PAYROL.B      Files in File Group PAYROL
PAYROL.C
A.ACCTS
A.INV
A.PROL        Files in File Group A
A.B.GO
A.B.INV
A.B.GO
A.B.INV       Files in File Group A.B
```

The saved files can be restored to the following systems:

- System/36 (RESTORE procedure or $COPY utility)
- AS/400 (Restore System/36 File (RSTS36F) command)

The SAVS36F command is intended for exchanging files with a System/36. For creating a backup version of a file, the AS/400 save commands (for example, Save Object (SAVOBJ) or Save Changed Object (SAVCHGOBJ)) should be used.

Restrictions:

1. The following authorities are required (normally only applies when running on a system using resource security):
   - *USE authority for this command.
   - *USE authority for the file or group of files specified in the FROMFILE parameter.
   - *USE authority for the library specified in the FROMLIB parameter.
   - *CHANGE authority to the file specified on the PHYFILE parameter if saving to an existing physical file.
   - *USE authority for the library specified in the PHYFILE parameter if saving to a physical file.

- *CHANGE authority for the library specified in the PHYFILE parameter if saving to a physical file and the file does not exist.
- *USE authority for the diskette device description object, *USE authority for device file QSYSDKT, in library QSYS if saving to diskette.
- *USE authority for the tape device description object, *USE authority for device file QSYSTAP, in library QSYS if saving to tape.
- *USE authority for the based-on physical file if saving a logical file.

2. All diskettes that are used for the save operation should be initialized using the INZDKT CL command or the equivalent System/36 environment function (INIT operator control language (OCL) procedure or $INIT SSP utility). For a two-sided diskette, use a sector size of 256 or 1024. For a one-sided diskette, use a sector size of 128 or 512. If tape is used, each tape volume used should have been initialized with standard labels using the INZTAP CL command or the equivalent System/36 environment function (TAPEINIT OCL procedure or $TINIT SSP utility). Use a density of 1600 bits per inch when initializing the tape.

   Note: If the tape or diskette has not been initialized as stated above, the System/36 will not be able to process the media.

3. Object-level and record-level functions, other than read operations, should not be attempted for a file being saved by SAVS36F. Concurrent activity against the file (for example, moving the file or adding or removing records) can cause:
   - For a save operation of a single file (FROMFILE(file-name)), the save operation will end with escape message CPF9826 because the file cannot be allocated.
   - For a save operation of multiple files (FROMFILE(*ALL or generic*-file-name)), the save function sends an inquiry message CPA2C6A because the file cannot be allocated. The message allows an ignore, retry and cancel response. The ignore response bypasses this file and attempts to save the next file selected.

4. When saving a single file to diskette, the diskette cannot already contain an active file with the same label and creation date as the new file to be created.

5. When saving multiple files to diskette, the diskette used for the save cannot contain any active files.

6. Not all physical and logical files can be saved with the SAVS36F command.
   - Only logical files created under the System/36 environment (for example, through the BLDINDEX OCL procedure) or through a DDM request from a System/36 system can be saved. These files are saved as System/36 alternative index files.
   - All physical files created under the System/36 environment (for example, through the BLDFILE OCL procedure) or through a DDM request from a System/36 system are saved using information stored within the AS/400 file description. These files are saved as System/36 sequential, direct, or indexed physical files.
   - Any physical files created by AS/400 commands or utilities can be saved as long as the record length is not greater than 4096. These files are saved as System/36 sequential files.

7. To generate a save format which can be processed by the System/36 RESTORE procedure, the following information is not saved:
   - If saving a logical file, only the description of the file is saved. The index (or access path) is not saved.
   - If saving an indexed (keyed) physical file, the data is saved but the index is not. The index will be rebuilt after the file is restored.

8. The following restrictions apply to naming standards:
   - When saving a single file, the specified name (FROMFILE parameter) must meet naming standards. If not, message CPF0001 is sent when the SAVS36F command is processed.
   - If a file name is found during a save operation of multiple files (FROMFILE(*ALL or generic*-file-name)) that does not meet the System/36 naming standards, diagnostic message CPF2C0E is sent and the file is not saved.

9. Multiple files (FROMFILE(*ALL) or FROMFILE(generic*-name)) cannot be saved to a physical file.

Parameters

Keyword	Description	Choices	Notes
FROMFILE	From file	*Generic name, name*, *ALL	Required, Positional 1
FROMLIB	From library	*Name*	Required, Positional 2
DEV	Device	Single values: *PHYFILE Other values (up to 4 repetitions): *Name*	Required, Positional 3
GROUP	File group qualifier	*Character value*, ***ALL**, *NONE	Optional
SET	Set identifier	*Character value*, **#SAVE**	Optional
TOLABEL	File label	*Character value*	Optional
SEQNBR	Sequence number	1-9999, ***END**	Optional
VOL	Volume identifier	Values (up to 50 repetitions): *Character value*, ***MOUNTED**	Optional
RETAIN	Retention period	0-999, **1**	Optional
ENDOPT	End of tape option	***REWIND**, *LEAVE, *UNLOAD	Optional
DTACPR	Data compression	*YES, ***NO**	Optional
PHYFILE	Physical file	*Qualified object name*	Optional
	Qualifier 1: Physical file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MBROPT	Data base member option	***NOREPLACE**, *REPLACE	Optional
CRTDATE	Creation date	*Date*, ***LAST**, *ALL	Optional

From file (FROMFILE)

Specifies the name of the file being saved. The file specified must exist in the library specified in the **From library** prompt (FROMLIB parameter).

This is a required parameter.

file-name
> Specify the name of a single file to save.

*ALL All files in the specified library are saved. To further describe which files are saved, use the **File group qualifier** prompt (GROUP parameter).

generic*-file-name
> Specify a generic name of a group of files in the specified library to save. All files with the same prefix as the generic name are saved. A generic name is specified as a character string of one or more characters, followed by an asterisk.

From library (FROMLIB)

Specifies which library contains the database files to be saved.

This is a required parameter.

Device (DEV)

Specifies the name of a diskette unit or the names of one or more tape devices. A maximum of four tape device names can be specified. If more than one tape device is used, enter the names of the devices in the order in which they are used. Each device name must be already known on the system by a device description.

This is a required parameter.

*PHYFILE

> A database physical file receives the copied file. The qualified name of the physical file must be specified on the **Physical file** prompt (PHYFILE parameter). *PHYFILE is not valid if *ALL or a generic*-file-name is specified for the **From file** prompt (FROMFILE parameter).

device-name

> Specifies the name of the diskette unit or the names of one or more tape devices used for the save operation.

File group qualifier (GROUP)

Specifies which file groups are to be saved. This parameter is valid only if *ALL is specified for the **From file** prompt (FROMFILE parameter).

*ALL   All files are saved.

*NONE

> No files that belong to a file group are saved. Only files that do not belong to file group are saved.

group-name

> Specify the name of a file group. All files that belong to that file group are saved. Files that do not belong to the specified file group are not saved. A group name can be up to 7 characters in length. The first character in the name must be an alphabetic character (A through Z, #, @, or $). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), quotation marks ("), question mark (?), asterisk (*), or blanks. The period, which indicates a file group name, must not be included as part of the group name.

Set identifier (SET)

Specifies the set identifier used to identify the entire set of files to be saved. This parameter is valid only if *ALL or a generic name is specified for the **From file** prompt (FROMFILE parameter).

#SAVE

> The default set identifier is used.

set-identifier

> Specify the set identifier used to identify the entire set of files to be saved. The set identifier can be up to 8 characters in length. The first character in the name must be an alphabetic character (A through Z, #, @, or $). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), quotation marks ("), question mark (?), asterisk (*), or blanks.

File label (TOLABEL)

Specifies the label value (eight characters maximum) given to the new diskette or tape file created by the save operation of a single file. If no value is specified, the value of the **From file** prompt (FROMFILE parameter) is used as the diskette or tape label.

This parameter is not allowed if *ALL or a generic name is specified for **From file** prompt (FROMFILE parameter).

Sequence number (SEQNBR)

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

*END   The specified file or files are saved starting after the last sequence number on the first tape (that is, this file follows all other files currently on the tape volume). If the first tape is full, an error message is issued and the operation ends. If the sequence number to be assigned to the specified file is greater than 9999, an error message is issued and the operation ends.

   If multiple files are saved, the next file is saved to a file after the first file that was saved, and so on. If the sequence number to be assigned ever exceeds 9999, an error message is issued and the operation ends.

file-sequence-number
   Specify the sequence number of the tape file that is used for the save operation. Valid values range from 1 through 9999.

   If this sequence number already exists on the tape volume, the tape label at that sequence number must match the TOLABEL parameter. The existing file at that sequence number is overwritten, and all subsequent files on the volume are not accessible after the save.

   If a new tape file is added to the tape, the sequence number must be one higher than the sequence number of the last tape file on that volume. No gaps are allowed in the series of sequence numbers.

   If multiple files are being saved, this sequence number is used for the first file. All remaining files are saved as if *END was specified on the parameter SEQNBR. If the sequence number to be assigned ever exceeds 9999, an error message is issued and the operation ends.

Volume identifier (VOL)

Specifies the volume identifiers of the tape volumes on which the object data is to be saved. The volumes must be placed in the device in the same order as the volume identifiers are specified for this parameter.

*MOUNTED
   The volume currently placed in the device is used.

volume-identifier
   Specify the volume identifiers of the tapes or diskettes used for the save operation. A maximum of 50 volume identifiers can be specified.

Retention period (RETAIN)

Specifies the retention period for the newly created tape or diskette file. The file is protected and cannot be written over until the day after the retention period ends.

1   A retention period of one day is used.

retention-period
>Specify the number of days the tape or diskette file should be kept. If a retention period of 999 is specified, the tape or diskette file becomes a permanent file.

End of tape option (ENDOPT)

Specifies, only when tape is used, what operation is automatically performed on the tape volume after the save operation ends. This parameter applies only to the last reel used.

*REWIND
>The tape is rewound, but not unloaded.

*LEAVE
>The tape is not rewound.

*UNLOAD
>The tape is automatically rewound and unloaded after the operation ends.

Data compression (DTACPR)

Specifies, when a diskette is used, whether the data is compressed into System/36 compatible format before it is written to the diskette. If the save command is operating while other jobs on the system are active and data compression is used, the overall system performance may be affected. This parameter is not valid if *PHYFILE or a tape device is specified on the **Device** prompt (DEV parameter).

The possible values are:

*NO   The data is not compressed before being written to the diskette.

*YES   The data is compressed before being written to the diskette.

Physical file (PHYFILE)

Specifies the name of the file that receives the copied file. If the specified file does not exist, it is created in the current library as a non-keyed, program-described file with a record length of 256. The copied records are put in the first member of the physical file. If the file has no members, a member is created using the system date.

The possible library values are:

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

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

library-name
> Specify the library where the file is located.

Data base member option (MBROPT)

Specifies whether the new records replace or are added to the existing records.

*NOREPLACE
> Specifies whether a file already exists by the name specified on the PHYFILE parameter in the specified library, an error message is sent and the data in that member is not replaced.

*REPLACE
> The PHYFILE member is cleared before copying the first record.

Creation date (CRTDATE)

Specifies, for a *date-differentiated* file (maintained by the System/36 environment), which instance (member) of the file or files is saved. A date-differentiated file has the same name as another file but it has a different file creation date. On this system, date-differentiated files are multiple-member physical files.

*LAST
> The most recently created member for the specified file or files are saved.

*ALL   All members in the date-differentiated file are saved. If the file being saved is not date differentiated, only the last member created in the file is saved. *ALL is valid only when *ALL or a generic file name is specified for **From file** prompt (FROMFILE parameter).

file-creation-date
> Specify the creation date of the date-differentiated file member to save. A file creation date is valid only when a single file is saved, or a file name is specified on the **From file** prompt (FROMFILE parameter).

Examples

Example 1: Saving a Single File

```
SAVS36F   FROMFILE(PETE)  FROMLIB(QS36F)  DEV(I1)
```

This command saves the file named PETE located in library QS36F. Assuming that I1 is the name of a diskette device description, the file is saved on the diskette placed in the diskette drive. The diskette file label is PETE (same as the FROMFILE name). If PETE is a date-differentiated physical file, the most recently created instance (member) of PETE is saved. The diskette file has a retention period of one day (the retention period ends at midnight of the following day).

Example 2: Saving a Single File

```
SAVS36F   FROMFILE(MSTRPAY)  FROMLIB(PAYLIB)  DEV(T1 T2)
          TOLABEL('PAY.MSTR')  RETAIN(999)
```

This command saves the file named MSTRPAY located in library PAYLIB. Assuming that T1 and T2 are tape devices, the file is copied to the tapes on devices T1 and T2. The tape file label is PAY.MSTR and the tape file is a permanent file. The last tape used for the save is rewound at the end of the save operation.

Example 3: Saving Multiple Files

```
SAVS36F   FROMFILE(*ALL)  FROMLIB(QS36F)  DEV(T1 T2)
          GROUP(*ALL)  SET(ALLFL)  RETAIN(999)
```

This command saves all database physical and logical files in library QS36F (including all files that belong to a file group). If any of the files are date-differentiated files, only the last member created in each file is saved. Assuming that T1 and T2 are tape devices, the files are copied to the tape volumes that are placed in tape drives T1 and T2. The label of the tape files created are the same as the names of the files that are saved. The first tape file created is located after the last sequence number on the tape. The remaining files are located after that first file. The tape files created are permanent. The last tape used for the save is rewound at the end of the save operation. The set identifier associated with this save all set is ALLFL.

Example 4: Saving Multiple Files

```
SAVS36F   FROMFILE(*ALL)  FROMLIB(QS36F)  DEV(T1 T2)
          GROUP(*NONE)  CRTDATE(*LAST) SET(NOGFL) RETAIN(999)
```

This command saves all database physical and logical files in library QS36F except those files that belong to a file group. If any of the files are date-differentiated files, only the last member created in each file is saved. Assuming that T1 and T2 are tape devices, the files are copied to the tape volumes that are placed in tape drives T1 and T2. The label of the tape files created is the same as the names of the files that are saved. The first tape file created is located after the last sequence number on the tape. The remaining files are located after that first file. The tape files created are permanent. The last tape used for the save is rewound at the end of the save operation. The set identifier associated with this save all set is NOGFL.

Example 5: Saving Multiple Files

```
SAVS36F   FROMFILE(*ALL)  FROMLIB(GRPLIB)  DEV(I1)
          GROUP(GRP)  CRTDATE(*ALL)
```

This command saves all database physical and logical files in library GRPLIB that belong to file group GRP (GRP.01, GRP.02, and so on). If any of the files are date-differentiated files, all members in the files are saved. Assuming that I1 is a diskette drive, the files are copied to the diskette that is placed in the diskette drive. The label of the diskette files created is the same as the names of the files that are saved. The diskette files expire after one day. The set identifier associated with this save all set is #SAVE.

Example 6: Saving Multiple Files

```
SAVS36F   FROMFILE(PAY*)  FROMLIB(PAYROLL)  DEV(I1)
          SET(PAYSET)  CRTDATE(*LAST)
          VOL(PAYDKT)  RETAIN(10)
```

This command saves all database physical and logical files in library PAYROLL whose names begin with the characters PAY (PAY.01, PAYRATE, and so on). If any of the files are date-differentiated files, only the last member created is saved. Assuming that I1 is a diskette drive, the files are copied to a diskette with a volume identifier of PAYDKT. The label of the diskette files created is the same as the names of the files that are saved. The diskette files expire after ten days. The set identifier associated with this save all set is PAYSET.

Error messages

*ESCAPE Messages

CPF2C4A
> Device &1 not correct for command.

CPF2C4B

    Duplicate device &1 specified in device name list.

CPF2C4C

    Diskette device &1 included in multiple device specification.

CPF2C4F

    Diskette format not correct for DTACPR(*YES).

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.

CPF2C5B

    Not all files were saved.

CPF2C5C

    Save operation ended before all files were saved.

CPF2C5D

    No files saved from library &1.

CPF2C5E

    Input file &1 in &2 not correct for command.

CPF2C5F

    Tape file sequence numbers beyond 9999 not allowed.

CPF2C50

    File description for file &1 is not available.

CPF2C51

    Member information for file &1 in library &2 is not available.

CPF2C52

    Error occurred during attempt to create file &1 in library &2.

CPF2C54

    FROMFILE name &1 too long to use for TOLABEL parameter.

CPF2C55

    TOLABEL parameter value &1 contains embedded blank(s).

CPF2C56

    Physical file name &1 too long.

CPF2C58

    Diskette format not acceptable for System/36.

CPF2C59

    FROMFILE name &1 too long.

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.

<u>***STATUS Messages**</u>

CPI2C13

Copying records from file &1 in library &2 member &3.

Top

Save S/36 Library Members (SAVS36LIBM)

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

The Save Library Members in System/36 Save/Restore Format (SAVS36LIBM) command creates a copy of source file members in a file on diskette or tape that can be restored on a System/36, or into a database physical file on this system that can be sent to a System/36. The saved member file is formatted like a record-mode LIBRFILE created on a System/36 using the System/36 FROMLIBR system operator control language (OCL) procedure (or the equivalent OCL use of the $MAINT SSP utility). On a System/36, the diskette or tape or disk file can be restored using the TOLIBR system OCL procedure (or the equivalent OCL use of the $MAINT SSP utility).

If a database physical file is specified using the **Physical file** prompt (PHYFILE parameter) but does not exist, it is created.

This command is intended only for exchanging source and procedure data with a System/36. It provides a simplified command interface for an AS/400 system customer who migrated from a System/36, but is not well-suited for backing up of AS/400 system source files. Use the AS/400 system CL commands (SAVOBJ or SAVCHGOBJ) for creating a backup copy of an AS/400 system source file or individual source file members.

Restrictions:

1. The following authorities are required when running on a system using resource security:
   - *USE authority for this command
   - *USE authority for the library specified in the FROMLIB parameter
   - *USE authority for file QS36SRC in the specified library if saving source library members
   - *USE authority for file QS36PRC in the specified library if saving procedure library members
   - *USE authority for the library specified in the PHYFILE parameter if saving to a physical file
   - *CHANGE authority for the library specified in the PHYFILE parameter if saving to a physical file and the file does not exist
   - *CHANGE and *OBJMGMT authority for that file if saving to a physical file with MBROPT(*ADD)
   - *ALL authority for the file if saving to a physical file with MBROPT(*REPLACE)
   - *USE authority for the diskette device description object, *USE authority for device file QSYSDKT in library QSYS if saving to diskette
   - *USE authority for the tape device description object and *USE authority for device file QSYSTAP in library QSYS if saving to tape
2. All diskettes that are used to save the members should be initialized using the INZDKT CL command or the equivalent System/36 environment function (INIT OCL procedure or $INIT SSP utility). For a two-sided diskette, use a sector size of 256 or 1024. For a one-sided diskette, use a sector size of 128 or 512. If tape is used, each tape volume used should first be initialized with standard labels using the INZTAP CL command or the equivalent System/36 environment function (TAPEINIT OCL procedure or $TINIT SSP utility). Use a density of 1600 bits per inch when initializing the tape.

   Note: If the tape or diskette has not been initialized as stated above, the System/36 will not be able to process the media.

   If a tape or diskette is used that has not been properly initialized, a message is sent to the system operator that allows the operator to cancel the save or initialize the volume and continue.

3. Object-level functions, other than read operations, should not be used for files QS36SRC and QS36PRC while members are being saved by SAVS36LIBM. If the necessary files cannot be allocated, no members are saved.

   Record-level functions, other than read operations, should not be used for members being saved. Concurrent activity against a member (for example, adding or removing records) can cause the member to be omitted from the save operation.

4. If saving a file to diskette, the diskettes used cannot contain an active file with the same name as the TOLABEL parameter value, because the AS/400 system does not allow duplicate diskette file labels.

5. Only members from source files QS36SRC (for *SRC members) and QS36PRC (for *PRC members) in the specified library can be saved using the SAVS36LIBM command. Only the member data is saved from the source file member (that is, the source sequence number and change date fields are not saved).

6. The specified member name or generic member name (FROMMBR parameter) must meet AS/400 system naming standards. When saving a member that has an extended name, the quotation mark characters are not stored as part of the member name in the output file. For example, member "A+B" would be saved as A+B.

Parameters

Keyword	Description	Choices	Notes
FROMMBR	From member	*Generic name, name*, *ALL	Required, Positional 1
FROMLIB	From library	*Name*	Required, Positional 2
DEV	Device	Single values: *PHYFILE Other values (up to 4 repetitions): *Name*	Required, Positional 3
SRCMBRS	S/36 source members	***ALL**, *SRC, *PRC	Optional
TOLABEL	File label	*Character value*	Optional
SEQNBR	Sequence number	1-9999, ***END**	Optional
VOL	Volume identifier	Values (up to 50 repetitions): *Character value*, ***MOUNTED**	Optional
RETAIN	Retention period	0-999, **1**	Optional
EXCHTYPE	Diskette file exchange type	***E**, *BASIC	Optional
ENDOPT	End of tape option	***REWIND**, *LEAVE, *UNLOAD	Optional
PHYFILE	Physical file	*Qualified object name*	Optional
	Qualifier 1: Physical file	*Name*	
	Qualifier 2: Library	*Name*, *CURLIB, ***LIBL**	
MBROPT	Data base member option	***REPLACE**, *ADD	Optional
RCDLEN	Record length	40-120, **120**	Optional

From member (FROMMBR)

Specifies the names of the members being saved.

This is a required parameter.

*ALL   All members of the member type specified on the **S/36 source members** prompt (SRCMBRS parameter) are saved.

member-name
>Specify the member name of the members to be saved.

generic-member-name*
>Specify the generic member name of the members to be saved. A generic name is a character string that contains one or more characters followed by an asterisk (*).

From library (FROMLIB)

Specifies which library contains the members being saved.

This is a required parameter.

Device (DEV)

Specifies the names of the devices used for the save operation. A maximum of four devices may be specified.

This is a required parameter.

*PHYFILE
>The output file is the database physical file specified by the **Physical file** prompt (PHYFILE parameter).

device-name
>Specify the name of the diskette unit or the names of one or more tape devices that are used for the save operation. If more than one tape device is used, type the names of the devices in the order in which they are used. A maximum of four tape device names can be specified.

S/36 source members (SRCMBRS)

Specifies which source member types (source and procedure members on System/36) are saved.

*ALL System/36 source and procedure members (from QS36SRC and QS36PRC) that match the member name specified on the **From member** prompt (FROMMBR parameter) are saved.

*SRC Only System/36 source members (from file QS36SRC) that match the member name specified on the **From member** prompt (FROMMBR parameter) are saved.

*PRC Only System/36 OCL procedure members (from file QS36PRC) that match the member name specified on the **From member** prompt (FROMMBR parameter) are saved.

File label (TOLABEL)

Specifies the label value (eight characters maximum) of the output diskette or tape file. If *PHYFILE is not specified on the **Device** prompt (DEV parameter), a value must be specified here.

Sequence number (SEQNBR)

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

*END   The specified members are saved starting after the last sequence number on the first tape (this file is put after all other files currently on the tape volume). If the first tape is full, an error message is issued and the operation ends. If the sequence number to be assigned to the specified file is greater than 9999, an error message is issued and the operation ends.

file-sequence-number
> Specify the sequence number of the tape file that is used when saving the specified members. Valid values range from 1 through 9999.
>
> If this sequence number already exists on the tape volume, the tape label at that sequence number must match the TOLABEL parameter value. The existing data file at that sequence number is overwritten, and all subsequent files on the volume are not accessible after the save operation.
>
> If a new tape file is added to the tape, the sequence number must be one greater than the sequence number of the last tape file on that volume. No gaps are allowed in the series of sequence numbers.

Volume identifier (VOL)

Specifies the volume identifiers of the tape volumes on which the object data is to be saved. The volumes must be placed in the device in the same order as the volume identifiers are specified for this parameter.

*MOUNTED
> The volume currently placed in the device is used.

volume-identifier
> Specify the volume identifiers of the tapes or diskettes used for saving the members. A maximum of 50 volume identifiers can be specified.

Retention period (RETAIN)

Specifies the retention period for the newly created tape or diskette file. The file is protected and cannot be written over until the day after the retention period ends.

1   A retention period of one day is used.

retention-period
> Specify the number of days the tape or diskette file should be kept. If a retention period of 999 is specified, the tape or diskette file becomes a permanent file.

Diskette file exchange type (EXCHTYPE)

Specifies the exchange type for the newly created diskette file.

*E   The default is to create an E-exchange diskette file. An E-exchange file is a system save file that can be restored on a System/36 using the TOLIBR procedure. It can also be restored to the AS/400 system using the Restore System/36 Library Members (RSTS36LIBM) command.

*BASIC

> The output diskette file is to be in basic exchange format. A basic exchange format file can be restored or copied to a System/34 or System/32 or any other system that supports the basic exchange diskette format.

End of tape option (ENDOPT)

Specifies, only when tape is used, what operation is automatically performed on the tape volume after the save operation ends. This parameter applies only to the last reel used.

*REWIND

> The tape is rewound, but not unloaded.

*LEAVE

> The tape is not rewound.

*UNLOAD

> The tape is automatically rewound and unloaded after the operation ends.

Physical file (PHYFILE)

Specifies the name of the physical file that receives the copied source file member data.

If a file by this name does not exist, it is created in the current library if a library name was not specified, as a non-keyed, program-described physical file with the record length specified by the **Logical record length** prompt (RCDLEN parameter). If a file already exists by this name, it is used as long as it is a non-keyed physical file with a record length in the range from 40 through 120. The copied records are put in the first member of the physical file.

The possible library values are:

*CURLIB

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

library-name

> Specify the library where the file is located.

Data base member option (MBROPT)

Specifies whether the new records replace or are added to the existing records.

*REPLACE

> The member is cleared before copying the first record.

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

Record length (RCDLEN)

Specifies the file record length (in bytes) used when copying the members.

120 This is the maximum record length allowed for System/36 source and procedure library members.

record-length
> Specify a record length in bytes.

Examples

Example 1: Saving Single Procedure Member

```
SAVS36LIBM   FROMMBR(XYZ1)  FROMLIB(JOHNSON)
             DEV(I1)  SRCMBRS(*PRC)  TOLABEL(XYZ1)
```

This command saves the single OCL procedure member XYZ1 (in source file QS36PRC) in library JOHNSON. Assuming I1 is a diskette device, the member is saved to diskette file XYZ1. The file length is 120 and the retention period is one day.

Example 2: Saving All Source and Procedure Members

```
SAVS36LIBM   FROMMBR(X*)  FROMLIB(ORDER)  DEV(*PHYFILE)
             PHYFILE(NETLIB/S36SRC) MBROPT(*ADD)
```

This command saves all source and procedure members (members in QS36SRC and QS36PRC) with names starting with the character 'X' in library ORDER. The members are saved into database physical file S36SRC in library NETLIB. The copied records are added after any records already in the file.

Error messages

*ESCAPE Messages

CPF2C4A
> Device &1 not correct for command.

CPF2C4B
> Duplicate device &1 specified in device name list.

CPF2C4C
> Diskette device &1 included in multiple device specification.

CPF2C43
> Saved &2 members from library &1, &3 members not saved.

CPF2C44
> No members saved from library &1.

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.

CPF2C5F

  Tape file sequence numbers beyond 9999 not allowed.

CPF2C50

  File description for file &1 is not available.

CPF2C51

  Member information for file &1 in library &2 is not available.

CPF2C52

  Error occurred during attempt to create file &1 in library &2.

CPF2C55

  TOLABEL parameter value &1 contains embedded blank(s).

CPF2C58

  Diskette format not acceptable for System/36.

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.

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.

<u>*STATUS Messages</u>

CPI2C13

  Copying records from file &1 in library &2 member &3.

Top

Save Save File Data (SAVSAVFDTA)

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

The Save Save File Data (SAVSAVFDTA) command saves the contents of a save file to tape or optical media. This command saves the save data in the save file to the device in a way that allows you to restore objects directly from the device.

A save file containing data created by the Save Licensed Program (SAVLICPGM) command cannot be saved by this command.

The information written on media by this command is similar to the data that was previously written to the save file by the save command that originally created the save file data. This includes the object descriptions, and object contents that existed when the original save operation was done.

This command uses only the save file and device description objects; it does not refer to or modify the description or contents of the objects included in the file save data. Thus, objects included in the save file are not locked during the running of this command, and the save history information (date, place, and time when each object was last saved) is not updated by this command for each object in the save file.

The description of the save file is not included in the save operation (unless it was included with the objects that were saved to create the save data in the file). In addition, this command does not update the save history information for the save file object, so the last save operation date, time, and place always identify the last save operation of the save file object description, not its contents.

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

Restrictions:

- You must have use (*USE) authority for the save file and *USE authority for the tape or optical device description.
- If a tape is used, it must have a standard label.
- The save file cannot be in use by a job running at the time the save operation occurs.

Top

Parameters

Keyword	Description	Choices	Notes
SAVF	Save file	*Qualified object name*	Required, Positional 1
	Qualifier 1: Save file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DEV	Device	Values (up to 4 repetitions): *Name*	Required, Positional 2
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional, Positional 3
SEQNBR	Sequence number	1-16777215, ***END**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
OPTFILE	Optical file	*Path name*, *****	Optional

Keyword	Description	Choices	Notes
USEOPTBLK	Use optimum block	*YES, *NO	Optional
CLEAR	Clear	*NONE, *ALL, *AFTER, *REPLACE	Optional
EXPDATE	File expiration date	Date, *PERM	Optional
COMPACT	Data compaction	*DEV, *NO	Optional
OUTPUT	Output	*NONE, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	Qualified object name	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
OUTMBR	Output member options	Element list	Optional
	Element 1: Member to receive output	Name, *FIRST	
	Element 2: Replace or add records	*REPLACE, *ADD	

Save file (SAVF)

Specifies the save file whose contents are to be saved.

Note: The save file must contain data written by a previous save command or an error message is sent, and its contents are not saved to media.

This is a required parameter.

Qualifier 1: Save file

name   Specify the name of the save file whose contents are to be saved.

Qualifier 2: Library

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

*CURLIB
> The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

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

Device (DEV)

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

This is a required parameter.

optical-device-name
> Specify the name of the optical device used for the save operation.

tape-media-library-device-name
> Specify the name of the tape media library device used for the save operation.

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

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED
> The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.
>
> **Note:** This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value
> Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215
> Specify the sequence number of the file to be used for the save operation.

End of media option (ENDOPT)

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

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

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

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

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

Top

Optical file (OPTFILE)

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

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

*
> The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/*'
> The system generates an optical file name in the specified directory of the optical volume.

Top

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

*YES The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:
- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

*NO The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Top

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE
>   None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*ALL   All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

*AFTER
>   All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*REPLACE
>   Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Top

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:
1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

*PERM
>   The file is protected permanently.

date   Specify the date when protection for the file ends.

Top

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

*DEV   Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

*NO   Device data compaction is not performed.

Top

Output (OUTPUT)

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

*NONE
No output listing is created.

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

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

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

File to receive output (OUTFILE)

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

Qualifier 1: File to receive output

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

Qualifier 2: Library

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

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

name Specify the name of the library to be searched.

Output member options (OUTMBR)

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

Element 1: Member to receive output

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

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

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

Element 2: Replace or add records

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

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

Top

Examples

```
SAVSAVFDTA   SAVF(ONLINE)   DEV(TAP01)   SEQNBR(1)   CLEAR(*ALL)
```

This command saves the contents of save file ONLINE to the first file on the tape volume on device TAP01. Files that have not ended on either the first tape volume or on subsequent volumes are overwritten without an inquiry message because CLEAR(*ALL) is specified.

Top

Error messages

*ESCAPE Messages

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.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

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

CPF3793
Machine or ASP storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

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

CPF3812

   Save file &1 in &2 in use.

CPF3814

   No objects from save file &1 in &2 saved to media.

CPF384E

   USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF388B

   Optical file path name not valid.

CPF5729

   Not able to allocate object &1.

CPF9812

   File &1 in library &2 not found.

CPF9822

   Not authorized to file &1 in library &2.

CPF9825

   Not authorized to device &1.

Top

Save Security Data (SAVSECDTA)

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

The Save Security Data (SAVSECDTA) command saves all security information without requiring a system in a restricted state. The SAVSECDTA command saves the same security information that is saved when a Save System (SAVSYS) command is run including the following:

- User Profiles
- Authorization Lists
- Authority Holders

Information saved with the SAVSYS or SAVSECDTA command can be restored using the Restore User Profiles (RSTUSRPRF) and Restore Authority (RSTAUT) commands.

Restrictions:

- You must have save system (*SAVSYS) special authority to run this command.
- Changes made to user profiles while the SAVSECDTA command is being run may not be reflected on the media, depending on when the changes occurred in relation to the save operation.
- Concurrent running of other SAVSECDTA commands is not allowed.
- If *YES is specified for the **Object pre-check (PRECHK)** parameter and a security object cannot be saved, the save operation ends.

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): *Name*	Required, Positional 1
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional
SEQNBR	Sequence number	1-16777215, ***END**	Optional
EXPDATE	File expiration date	*Date*, ***PERM**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
USEOPTBLK	Use optimum block	***YES**, *NO	Optional
SAVF	Save file	*Qualified object name*	Optional
	Qualifier 1: Save file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OPTFILE	Optical file	*Path name*, *****	Optional
ASPDEV	ASP device	*Name*, ***ALLAVL**, *, *SYSBAS, *CURASPGRP	Optional
CLEAR	Clear	***NONE**, *ALL, *AFTER, *REPLACE	Optional
PRECHK	Object pre-check	***NO**, *YES	Optional
DTACPR	Data compression	***DEV**, *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	***DEV**, *NO	Optional
OUTPUT	Output	***NONE**, *PRINT, *OUTFILE	Optional

Keyword	Description	Choices	Notes
OUTFILE	File to receive output	*Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	

Device (DEV)

Specifies the name of the device used for the save operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

*SAVF The save operation is done using the save file specified for the **Save file (SAVF)** parameter.

Other values

optical-device-name
> Specify the name of the optical device used for the save operation.

tape-media-library-device-name
> Specify the name of the tape media library device used for the save operation.

tape-device-name
> Specify the names of one or more tape devices used for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED
> The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

> **Note:** This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value
> Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215
> Specify the sequence number of the file to be used for the save operation.

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.

2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

*PERM
> The file is protected permanently.

date Specify the date when protection for the file ends.

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the save operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

Note: This parameter is valid only if a tape or optical device name is specified for the DEV parameter. For optical devices, *UNLOAD is the only special value supported, *REWIND and *LEAVE will be ignored.

*REWIND
> The tape is automatically rewound, but not unloaded, after the operation has ended.

*LEAVE
> The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

*UNLOAD
> The tape is automatically rewound and unloaded after the operation ends. Some optical devices will eject the volume after the operation ends.

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

*YES   The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

  • Performance may improve.
  • The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
  • The value for the DTACPR parameter is ignored.

*NO   The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name   Specify the name of save file to be used.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
        The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

name   Specify the name of the library where the save file is located.

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the save operation, beginning with the root directory of the volume.

For more information on specifying path names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

*   The system generates an optical file name in the root directory of the optical volume.

'*optical-directory-path-name*/*'
        The system generates an optical file name in the specified directory of the optical volume.

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device from which private authorities are to be saved.

*ALLAVL
> The private authorities from the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and all available independent ASPs are saved.

*   The private authorities 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 saved.

*SYSBAS
> The private authorities from the system ASP and all basic user ASPs are saved.

*CURASPGRP
> If the current thread has an ASP group, the private authorities from all independent ASPs in the ASP group are saved.

name   Specify the ASP device name from which private authorities are to be saved.

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE
> None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*ALL   All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

*AFTER
> All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the

save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

> **Note:** The *AFTER value is not valid for save files.

*REPLACE
> Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Object pre-check (PRECHK)

Specifies whether the save operation should end if all objects specified by this command do not satisfy the following conditions of the save operation: (1) the objects exist, (2) they were not previously found to be damaged, (3) they are not locked by another job, and (4) the requester of the save operation has authority to save the objects.

*NO The save operation continues, saving only those objects that can be saved.

*YES If, after all specified objects are checked, one or more objects cannot be saved, the save operation ends before any data is written.

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

*DEV If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

*NO No data compression is performed.

*YES If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

*LOW If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

> **Note:** This value is not valid for tape.

*MEDIUM
> If the save operation is to a save file or optical, software data compression is performed with the

TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

Note: This value is not valid for tape.

*HIGH

If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

Note: This value is not valid for tape.

Top

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

*DEV Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

Note: If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

*NO Device data compaction is not performed.

Top

Output (OUTPUT)

Specifies whether a list with information about the saved objects is created. The information can be printed with the job's spooled output or directed to a database file.

*NONE

No output listing is created.

*PRINT

The output is printed with the job's spooled output.

*OUTFILE

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the **File to receive output (OUTFILE)** parameter when OUTPUT(*OUTFILE) is specified.

Top

Save Security Data (SAVSECDTA) **583**

File to receive output (OUTFILE)

Specifies the database file to which the information is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QASAVOBJ in QSYS with the format name QSRSAV as a model.

Qualifier 1: File to receive output

name   Specify the name of the database file to which output from the command is directed. If this file does not exist, it is created in the specified library.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
         The current library for the thread is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

name   Specify the name of the library to be searched.

Top

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

*FIRST
         The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name   Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

         If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

*REPLACE
         The existing records in the specified database file member are replaced by the new records.

*ADD   The new records are added to the existing information in the specified database file member.

Top

Examples

Example 1: Automatically Clearing Uncleared Tapes

```
SAVSECDTA   DEV(TAP01)   CLEAR(*ALL)
```

This command saves the security information, including user profiles, authorization lists, authority holders. They are saved on the TAP01 tape drive. CLEAR(*ALL) automatically clears all uncleared tapes when they are encountered.

Example 2: Sending Message When Storage Capacity Exceeded

```
SAVSECDTA   DEV(TAP01)   VOL(ABC)
```

This command saves the security information on the TAP01 tape drive, starting on the tape volume labeled ABC. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be put on the TAP01 tape drive is shown to the operator.

Error messages

*ESCAPE Messages

CPF2206
> User needs authority to do requested function on object.

CPF222E
> &1 special authority is required.

CPF370A
> Not all security objects saved to save file &3.

CPF3709
> Tape devices do not support same densities.

CPF3727
> Duplicate device &1 specified on device name list.

CPF3728
> Device &1 specified with other devices.

CPF3731
> Cannot use &2 &1 in library &3.

CPF3733
> &2 &1 in &3 previously damaged.

CPF3735
> Storage limit exceeded for user profile &1.

CPF3737
> Save and restore data area &1 not found.

CPF3738
> Device &1 used for save or restore is damaged.

CPF3767
> Device &1 not found.

CPF3768
> Device &1 not valid for command.

CPF3782
> File &1 in &2 not a save file.

CPF3793
> Machine or ASP storage limit reached.

CPF3794
> Save or restore operation ended unsuccessfully.

CPF3812
> Save file &1 in &2 in use.

CPF384E

USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF388B

Optical file path name not valid.

CPF3893

Not all security objects saved.

CPF3894

Cancel reply received for message &1.

CPF38A4

ASP device &1 not correct.

CPF5729

Not able to allocate object &1.

CPF9809

Library &1 cannot be accessed.

CPF9812

File &1 in library &2 not found.

CPF9814

Device &1 not found.

CPF9833

*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9850

Override of printer file &1 not allowed.

CPF9851

Overflow value for file &1 in &2 too small.

CPF9860

Error occurred during output file processing.

CPFB8ED

Device description &1 not correct for operation.

Top

Save Storage (SAVSTG)

Where allowed to run: Interactive environments (*INTERACT *IPGM *IREXX *EXEC)	Parameters Examples
Threadsafe: No	Error messages

The Save Storage (SAVSTG) command saves a copy of the licensed internal code and the contents of auxiliary storage (except unused space and temporary objects) to tape. This function is intended for disaster recovery backup. Individual libraries or objects cannot be restored from a save storage tape.

This command issues the PWRDWNSYS (Power Down System) command with OPTION(*IMMED) and RESTART(*YES) specified. While the system is powered down, a dedicated service tool (DST) that saves all system storage is called. At that time, a standard labeled tape volume must be placed in the tape device. Additional volumes are requested as needed. Hardware data compression is used if it is supported by the tape device. After the save system storage function is complete, an initial program load (IPL) takes place.

During the IPL after a save storage operation, a completion message is sent to the system operator message queue. The save history information for the data area QSAVSTG in library QSYS is updated with the date and time when the system storage data was saved. To show the information in this data area, use the Display Object Description (DSPOBJD) command with DETAIL(*FULL).

The restore storage operation is done using the appropriate option on the DST menu. During the IPL, after a restore storage operation, a completion message is sent to the system operator message queue, and the last restore date and time history information in the QSAVSTG data area is updated with the current date and time. In addition, the data portion of the QSAVSTG data area is updated with the date of the save storage tape used in the restore system storage operation.

Note: Because media errors cause the save operation to start over from the last tape volume, use of this command is recommended for smaller systems only.

Restrictions:
- You must have save system (*SAVSYS) special authority to run this command.
- The system cannot be running any other jobs; run the End Subsystem (ENDSBS) specifying SBS(*ALL) or run the End System (ENDSYS) command to end all other jobs except the job running the SAVSTG command.
- Tapes created using this command that will be used for installation should be initialized with a density that is supported by the current alternate IPL device. If this is not done, the current IPL tape will have to be changed to a tape device that supports the density of the created SAVSTG tapes before installation can begin.
- Tapes created using the SAVSTG command should not be used for automatic installation.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Tape device	*Name*	Required, Positional 1
EXPDATE	File expiration date	*Date*, ***PERM**	Optional

Keyword	Description	Choices	Notes
CLEAR	Clear	*NONE, *ALL	Optional

Tape device (DEV)

Specifies the tape device that is used for the save storage operation.

This is a required parameter.

name   Specify the name of the tape device to be used.

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

*PERM
>   The tape files are protected permanently.

date   Specify the date when protection for the tape files ends. The date must be specified in the job date format.

Clear (CLEAR)

Specifies whether uncleared tapes encountered during the save operation are automatically cleared. An uncleared tape is one containing a file with an expiration date later than the date of the save operation, which includes files protected permanently using EXPDATE(*PERM).

Note: This parameter does not control initializing the tapes. The tapes used to perform the save operation should be initialized to a standard label format before the save command is issued. You can use the Initialize Tape (INZTAP) command and specify a value for the NEWVOL parameter to initialize a tape to a standard label format. If a tape volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the tape volume.

*NONE
>   None of the uncleared tapes encountered during the save operation are automatically cleared. If the save operation cannot proceed because an uncleared tape is encountered, an inquiry message is sent to the operator, who is allowed to end the save operation or to specify that the selected tape be cleared so the operation can continue.

*ALL   All uncleared tapes encountered during the save operation are automatically cleared.

Examples

Example 1: Specifying Expiration Date

```
SAVSTG  DEV(TAP01)  EXPDATE(122290)  CLEAR(*ALL)
```

This command saves the system storage on the tape put on the TAP01 tape drive. Each uncleared tape is cleared automatically. The tape files written are protected and cannot be overwritten until December 22, 1990.

Example 2: Saving System Storage

```
SAVSTG   DEV(TAP02)
```

The system storage is saved on tape drive TAP02. CLEAR was not specified, so uncleared tapes encountered during the save operation cause an inquiry message to be sent to the operator, who either ends the save operation or specifies that the currently selected tape be cleared so the operation can continue. Because EXPDATE also was not specified, the tape files being written are protected permanently.

Top

Error messages

*ESCAPE Messages

CPF2206
> User needs authority to do requested function on object.

CPF222E
> &1 special authority is required.

CPF376A
> System must not be in manual mode.

CPF3767
> Device &1 not found.

CPF3768
> Device &1 not valid for command.

CPF3785
> Not all subsystems ended.

Top

Save System (SAVSYS)

The Save System (SAVSYS) command saves a copy of the Licensed Internal Code and the QSYS library in a format compatible with the installation of the operating system. It does not save objects from any other library. In addition, it saves security and configuration objects that can also be saved using the Save Security Data (SAVSECDTA) and Save Configuration (SAVCFG) commands.

To save the system data on offline storage, the system writes a copy of the objects onto the media. The libraries and objects are not affected on the system. This command cannot be used to free any space occupied by these objects. The history information for the data area QSAVUSRPRF in QSYS is updated with the date, time, and place where user profiles are saved. The history information for the data area QSAVSYS in QSYS is updated with the date, time, and place where the system is saved. The history information for the data area QSAVCFG in QSYS is updated with the date, time, and place where configuration objects are saved. The history information is not updated for the individual objects. To display the information in these data areas, run the Display Object Description (DSPOBJD) command, and specify DETAIL(*FULL). Save the information from the display of QSAVUSRPRF for the location where the user profiles are saved.

When using this command, it is important to use the device on the system that is defined as the initial program load (IPL) device. The IPL device was defined by the service representative when the system was installed. If an IPL device is not used when using this command, then the system cannot be restored using the SAVSYS media (if densities or media types are incompatible).

Restrictions:

1. You must have save system (*SAVSYS) special authority to run this command
2. All subsystems must be inactive before the SAVSYS command can be specified. The End System (ENDSYS) or End Subsystem (ENDSBS) command can be used to make the subsystems inactive. You must have job control (*JOBCTL) special authority to use the ENDSYS or the ENDSBS command.
3. Tapes created using this command that will be used for installation should be initialized with a density that is supported by the current IPL tape unit. If this is not done, the current IPL tape will have to be changed to a tape device that supports the density of the created SAVSYS tapes before installation begins.
4. Tapes created using the SAVSYS command should not be used for automatic installation.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Values (up to 4 repetitions): *Name*	Required, Positional 1
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional, Positional 2
EXPDATE	File expiration date	*Date*, ***PERM**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
USEOPTBLK	Use optimum block	***YES**, *NO	Optional

Keyword	Description	Choices	Notes
OMIT	Omit	Single values: ***NONE** Other values (up to 2 repetitions): *CFG, *SECDTA	Optional
OUTPUT	Output	***NONE**, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	*Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	
ASPDEV	ASP device	*Name*, ***ALLAVL**, *, *SYSBAS, *CURASPGRP	Optional
CLEAR	Clear	***NONE**, *ALL, *AFTER, *REPLACE	Optional
DTACPR	Data compression	***DEV**, *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	***DEV**, *NO	Optional

Device (DEV)

Specifies the name of the device used for the save operation. The device name must already be known on the system by a device description.

This is a required parameter.

optical-device-name
> Specify the name of the optical device used for the save operation.

tape-media-library-device-name
> Specify the name of the tape media library device used for the save operation.

tape-device-name
> Specify the names of one or more tape devices used for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED
> The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value
> Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

*PERM
> The files are protected permanently.

date   Specify the date when protection for the files ends.

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the save operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

Note: This parameter is valid only if a tape or optical device name is specified for the DEV parameter. For optical devices, *UNLOAD is the only special value supported, *REWIND and *LEAVE will be ignored.

*REWIND
> The tape is automatically rewound, but not unloaded, after the operation has ended.

*LEAVE
> The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

*UNLOAD
> The tape is automatically rewound and unloaded after the operation ends. Some optical devices will eject the volume after the operation ends.

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

*YES   The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:
- Performance may improve.

- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.

- The value for the DTACPR parameter is ignored.

*NO The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Note: Not all files will be saved with optimum block size.

Omit (OMIT)

Specifies what data to omit from the save system operation.

Single values

*NONE
All of the security data and configuration data are included in the save system operation.

Other values (up to 2 repetitions)

*CFG All of the configuration data is omitted from the save system operation. You can use the SAVCFG (Save Configuration) command to save just the configuration object information.

*SECDTA
All of the security data is omitted from the save system operation. You can use the SAVSECDTA (Save Security Data) command to save just the system security data.

Output (OUTPUT)

Specifies whether a list with information about the saved objects is created. The information can be printed with the job's spooled output or directed to a database file.

*NONE
No output listing is created.

*PRINT
The output is printed with the job's spooled output.

*OUTFILE
The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the **File to receive output (OUTFILE)** parameter when OUTPUT(*OUTFILE) is specified.

File to receive output (OUTFILE)

Specifies the database file to which the information is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QASAVOBJ in QSYS with the format name QSRSAV as a model.

Qualifier 1: File to receive output

name   Specify the name of the database file to which output from the command is directed. If this file does not exist, it is created in the specified library.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
        The current library for the thread is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

name   Specify the name of the library to be searched.

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

*FIRST
        The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name   Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

        If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

*REPLACE
        The existing records in the specified database file member are replaced by the new records.

*ADD   The new records are added to the existing information in the specified database file member.

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device from which private authorities are to be saved.

*ALLAVL
        The private authorities from the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and all available independent ASPs are saved.

* The private authorities 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 saved.

*SYSBAS
  The private authorities from the system ASP and all basic user ASPs are saved.

*CURASPGRP
  If the current thread has an ASP group, the private authorities from all independent ASPs in the ASP group are saved.

name Specify the ASP device name from which private authorities are to be saved.

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. Clearing active data removes all files from the volume. Replacing active data on optical media initializes the first volume, and replaces only the optical files created by this operation on any volumes after the first volume.

Notes:
1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE
  None of the media is automatically cleared. If the save operation encounters active data on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters active data after the first volume, an inquiry message is sent, allowing the operator to either end the save operation, clear the tape, or replace the optical file.

*ALL All of the media is automatically cleared.

*AFTER
  All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*REPLACE
  Active data on the media is automatically replaced. The first optical volume is initialized. Other optical volumes are not initialized. Tapes are automatically cleared in the same way as the *ALL value.

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

<u>*DEV</u>   If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

***NO**   No data compression is performed.

***YES**   If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

***LOW**   If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

> **Note:** This value is not valid for tape.

***MEDIUM**
> If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

> **Note:** This value is not valid for tape.

***HIGH**
> If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

> **Note:** This value is not valid for tape.

<div align="right">Top</div>

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

<u>*DEV</u>   Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

> **Note:** If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

> If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

***NO**   Device data compaction is not performed.

<div align="right">Top</div>

Examples

Example 1: Tapes Cleared Automatically

```
SAVSYS   DEV(TAP01)  CLEAR(*ALL)
```

This command saves licensed internal code, system objects, all user profiles (including private authority for objects), and all line, controller, and device descriptions. They are saved on the tape put on the TAP01 tape drive. Each uncleared tape is automatically cleared when it is encountered, and the save operation continues without operator intervention.

Example 2: Operating Receives Message of Exceeded Storage Capacity

```
SAVSYS   DEV(TAP01)  VOL(ABCDE)
```

The system data is saved on the TAP01 tape drive, starting on the tape volume labeled ABCDE. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be put on the TAP01 tape drive is shown to the operator.

Example 3: Saving Data on Two Tape Drives in Alternating Order

```
SAVSYS   DEV(TAP01 TAP02)
```

The system data is saved on tape drives TAP01 and TAP02 in alternating order. If the save operation exceeds the storage capacity of two tapes, a message requesting that another volume be put on TAP01 is shown to the operator. The tapes are rewound at the completion of the save system operation.

Error messages

*ESCAPE Messages

CPF2206
> User needs authority to do requested function on object.

CPF222E
> &1 special authority is required.

CPF3703
> &2 &1 in &3 not saved.

CPF3709
> Tape devices do not support same densities.

CPF372B
> Not all objects were saved.

CPF3727
> Duplicate device &1 specified on device name list.

CPF3728
> Device &1 specified with other devices.

CPF3733
> &2 &1 in &3 previously damaged.

CPF3735
> Storage limit exceeded for user profile &1.

CPF3738
> Device &1 used for save or restore is damaged.

CPF3767

Device &1 not found.

CPF3768

Device &1 not valid for command.

CPF3772

SAVSYS completed. One or more objects not saved.

CPF3785

Not all subsystems ended.

CPF3793

Machine or ASP storage limit reached.

CPF3794

Save or restore operation ended unsuccessfully.

CPF3797

Objects from library &3 not saved. Save limit exceeded.

CPF3798

Installation &2 &1 in &3 not found.

CPF384E

USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3873

Licensed program &1 option &2 release &4 not saved.

CPF388B

Optical file path name not valid.

CPF3894

Cancel reply received for message &1.

CPF38A4

ASP device &1 not correct.

CPF9814

Device &1 not found.

CPF9833

*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9850

Override of printer file &1 not allowed.

CPF9851

Overflow value for file &1 in &2 too small.

CPF9860

Error occurred during output file processing.

CPFB8ED

Device description &1 not correct for operation.

Save System Information (SAVSYSINF)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Save System Information (SAVSYSINF) command saves a subset of system data and objects saved by the Save System (SAVSYS) command.

SAVSYSINF is not to be considered a replacement for the SAVSYS command and is not to be used for system upgrades or migrations.

Objects saved from QSYS include:
- *JOBD
- *JOBQ
- *EDTD
- *JRN
- *SBSD
- *CLS
- *MSGQ
- *TBL
- *IGCTBL
- *DTAARA
- *CMD objects changed since the last SAVSYS
- *MSGF objects changed since the last SAVSYS

Additional items saved include:
- System reply list
- Service attributes
- Environment variables
- Certain system values
- Network attributes
- PTFs applied since the last SAVSYS

Some items NOT saved as part of SAVSYSINF command:
- Licensed Internal Code
- QSYS library
- Security objects (use the SAVSECDTA command)
- Configuration objects (use the SAVCFG command)
- User profiles (use the SAVSECDTA command)
- System values related to date/time
- System values which cannot be changed. For system values which can be changed, refer to the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter
- QPWDLVL (Password level) system value

Note: The SAVSYSINF command issues several save commands for saving objects. Parameters may or may not be used for all save commands.

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): *Name*	Required, Positional 1
VOL	Volume identifier	Single values: ***MOUNTED** Other values (up to 75 repetitions): *Character value*	Optional
SEQNBR	Sequence number	1-16777215, ***END**	Optional
EXPDATE	File expiration date	*Date*, ***PERM**	Optional
ENDOPT	End of media option	***REWIND**, *LEAVE, *UNLOAD	Optional
SAVF	Save file	*Qualified object name*	Optional
	Qualifier 1: Save file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OPTFILE	Optical file	*Path name*, *****	Optional
USEOPTBLK	Use optimum block	***YES**, *NO	Optional
CLEAR	Clear	***NONE**, *ALL, *AFTER, *REPLACE	Optional
DTACPR	Data compression	***DEV**, *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	***DEV**, *NO	Optional
OUTPUT	Output	***NONE**, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	*Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	

Device (DEV)

Specifies the name of the device used for the save operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

*SAVF The save operation is done using the save file specified for the **Save file (SAVF)** parameter.

Other values

optical-device-name
> Specify the name of the optical device used for the save operation.

tape-media-library-device-name
> Specify the name of the tape media library device used for the save operation.

tape-device-name
> Specify the names of one or more tape devices used for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED
> The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.
>
> **Note:** This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value
> Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215
> Specify the sequence number of the file to be used for the save operation.

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

*PERM
> The file is protected permanently.

date Specify the date when protection for the file ends.

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the save operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

Note: This parameter is valid only if a tape or optical device name is specified for the DEV parameter. For optical devices, *UNLOAD is the only special value supported, *REWIND and *LEAVE will be ignored.

*REWIND

>The tape is automatically rewound, but not unloaded, after the operation has ended.

*LEAVE

>The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

*UNLOAD

>The tape is automatically rewound and unloaded after the operation ends. Some optical devices will eject the volume after the operation ends.

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

Qualifier 2: Library

*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB

>The current library for the thread is used to locate the save file. If no current library entry exists in the library list, the QGPL library is used.

name Specify the name of the library where the save file is located.

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the save operation, beginning with the root directory of the volume.

For more information on specifying path names, refer to ″Object naming rules″ in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

* The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/'*
> The system generates an optical file name in the specified directory of the optical volume.

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

*YES The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

*NO The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE
> None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

*ALL All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

*AFTER

All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

*REPLACE

Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Top

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

*DEV If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

*NO No data compression is performed.

*YES If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

*LOW If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

Note: This value is not valid for tape.

*MEDIUM

If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

Note: This value is not valid for tape.

*HIGH

If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

Note: This value is not valid for tape.

Top

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

*DEV Device data compaction is performed if the data is saved to tape and all tape devices specified
for the **Device (DEV)** parameter support the compaction feature.

> **Note:** If *DEV is specified for both the **Data compression (DTACPR)** parameter and this
> parameter, only device data compaction is performed if device data compaction is supported on
> the device. Otherwise, data compression is performed if supported on the device.

> If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both
> device data compaction and device data compression are performed if supported on the device.

*NO Device data compaction is not performed.

Output (OUTPUT)

Specifies whether a list with information about the saved objects is created. The information can be
printed with the job's spooled output or directed to a database file.

*NONE
No output listing is created.

*PRINT
The output is printed with the job's spooled output.

*OUTFILE
The output is directed to the database file specified for the **File to receive output (OUTFILE)**
parameter.

> **Note:** You must specify a database file name for the **File to receive output (OUTFILE)** parameter
> when OUTPUT(*OUTFILE) is specified.

File to receive output (OUTFILE)

Specifies the database file to which the information is directed when *OUTFILE is specified for the
Output (OUTPUT) parameter. If the file does not exist, this command creates a database file in the
specified library. If a new file is created, the system uses QASAVOBJ in QSYS with the format name
QSRSAV as a model.

Qualifier 1: File to receive output

name   Specify the name of the database file to which output from the command is directed. If this file
does not exist, it is created in the specified library.

Qualifier 2: Library

*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
The current library for the thread is used to locate the file. If no library is specified as the current
library for the job, the QGPL library is used.

name   Specify the name of the library to be searched.

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

*FIRST

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name   Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

*REPLACE

The existing records in the specified database file member are replaced by the new records.

*ADD   The new records are added to the existing information in the specified database file member.

Examples

Example 1: Tapes Cleared Automatically

```
SAVSYSINF DEV(TAP01)  CLEAR(*ALL)
```

This command saves the system information on the tape put on the TAP01 tape drive. Each uncleared tape is automatically cleared when it is encountered, and the save operation continues without operator intervention.

Example 2: Save File Cleared Automatically and Output Generated to a File

```
SAVSYSINF DEV(*SAVF) SAVF(QGPL/SAVF) CLEAR(*ALL)
          OUTPUT(*OUTFILE) OUTFILE(QGPL/OUTPUT)
```

This command saves the system information to the save file named SAVF in library QGPL. The save file will be cleared automatically. Information about what was saved will be written to the first member of the file named OUTPUT in library QGPL; the file and member will be created if they do not exist.

Error messages

*ESCAPE Messages

CPF38A7

SAVSYSINF completed. One or more objects not saved.

CPD37AD

Save file not found for PTF &1-&2 &3.

Submit Data Base Jobs (SBMDBJOB)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Submit Database Jobs (SBMDBJOB) command allows you to submit jobs to job queues so they can be run as batch jobs. The input stream is read either from a physical database file or from a logical database file that has a single-record format. This command allows you to specify the name of this database file and its member, the name of the job queue to be used, and to decide whether jobs being submitted can be displayed by the Work with Submitted Jobs (WRKSBMJOB) command

A Submit Database Jobs operation reads the file once and ends when the end-of-file is read or when an End Input (ENDINP) command is encountered. The ENDINP command (a delimiter) is not recognized if it is within an inline file that ends with characters that are not default ending characters (as specified in the ENDCHAR parameter of the Data (DATA) command). The SBMDBJOB operation can be canceled either by canceling the request from the system request menu or by canceling the job in which the process is running.

In contrast to a spool reader started with the Start Database Reader (STRDBRDR) command, the SBMDBJOB command operates in the same process as the requesting function and does not do syntax checking on the input stream.

Restriction: The specified database file either must consist of single-field records and must have an arrival sequence access path, or it must be a standard database source file.

Parameters

Keyword	Description	Choices	Notes
FILE	Data base file	*Qualified object name*	Required, Positional 1
	Qualifier 1: Data base file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MBR	Member	*Name*, ***FIRST**	Optional, Positional 2
JOBQ	Job queue	*Qualified object name*	Optional, Positional 3
	Qualifier 1: Job queue	*Name*, **QBATCH**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DSPSBMJOB	Show on submitted job list	***YES**, *NO	Optional

Data base file (FILE)

Specifies the database file from which the input stream is read.

This is a required parameter.

Qualifier 1: Data base file

name Specify the name of the database file that contains the input stream that you want to read.

Qualifier 2: Library

*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
>   The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

name Specify the name of the library where the file is located.

Member (MBR)

Specifies the member in the specified file that contains the input stream to be read.

*FIRST
>   No member name is specified; the first member in the file is used.

name Specify the name of the member that contains the input stream to read.

Job queue (JOBQ)

Specifies the job queue on which the job entries are placed. A job entry is placed on this queue for each job in the input stream that has JOBQ(*RDR) specified on the Batch Job (BCHJOB) command. If *RDR is not specified on the BCHJOB command, the job queue specified on the BCHJOB command or in the job description is used. (The job queue for each job in the input stream can be different.) This parameter is valid only if ACTION(*SUBMIT) is specified in the existing network job entry or in a subsequent Change Network Job Entry (CHGNETJOBE) command.

Note: If both the user identified in the job description of the job being read and the user processing the Submit Database Job (SBMDBJOB) command are not authorized to the job queue on which the job should be placed, the job ends and a diagnostic message is placed in the job log. The input stream, continues to be processed, starting with the next job. If either user is authorized to the job queue, the job runs without error.

Qualifier 1: Job queue

QBATCH
>   The job entry is to be placed on the QBATCH job queue, which is the default job queue. This will happen if *RDR is specified on the **Job queue (JOBQ)** parameter of the Batch Job (BCHJOB) command. The Batch Job (BCHJOB) command is contained in the input stream itself.

name Specify the name of the job queue to which each job in the input stream will be sent. This will happen if *RDR is specified on the JOBQ parameter of the Batch Job (BCHJOB) command.

Qualifier 2: Library

*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
>   The current library for the job is used to locate the job queue.

name Specify the name of the library where the job queue is located.

Show on submitted job list (DSPSBMJOB)

Specifies whether the jobs being submitted are displayed on the submitted jobs display. If *YES is specified here, these submitted jobs can be displayed when the Work with Submitted Jobs (WRKSBMJOB) command is used.

*YES This job can be displayed by the Work with Submitted Jobs (WRKSBMJOB) command.

*NO This job is not displayed on any display produced by the Work with Submitted Jobs (WRKSBMJOB) command.

Examples

```
SBMDBJOB   FILE(QGPL/BILLING)
```

This command submits jobs using input from the database file named BILLING, which is in the QGPL library. The first member in the BILLING file contains the input stream to be processed. The default system-supplied job queue QBATCH is used.

Error messages

*ESCAPE Messages

CPF1751
> Error while processing job &3/&2/&1.

CPF1754
> File &1 in library &2 not database file or DDM file.

CPF1760
> Submit jobs command not allowed.

CPF2207
> Not authorized to use object &1 in library &3 type *&2.

CPF3307
> Job queue &1 in &2 not found.

CPF3330
> Necessary resource not available.

CPF3363
> Message queue &1 in library &2 not found.

CPF9802
> Not authorized to object &2 in &3.

CPF9812
> File &1 in library &2 not found.

CPF9815
> Member &5 file &2 in library &3 not found.

CPF9845
> Error occurred while opening file &1.

CPF9846

   Error while processing file &1 in library &2.

<u>*STATUS Messages</u>

CPF1762

   Reading job &3/&2/&1.

Top

Submit Finance Job (SBMFNCJOB)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Submit Finance Job (SBMFNCJOB) command submits a batch job that lets your finance application programs communicate with your 4701 or 4702 controller application programs.

Use the SBMFNCJOB command only if:

- Communicating with a 4701 or 4702 control unit
- A device table and a program table have been defined using the Work with Device Table (WRKDEVTBL) and Work with Program Table (WRKPGMTBL) commands; defining a user table using the Work with User Table (WRKUSRTBL) command is optional
- The user's 4701 or 4702 control unit application program sends data (transactions) first and expects a response
- The user's 4701 or 4702 control unit application program passes data in the proper format

Restriction: This command is shipped with public *EXCLUDE authority.

Parameters

Keyword	Description	Choices	Notes
DEVTBL	Device table	*Name*	Required, Positional 1
PGMTBL	Program table	*Name*	Required, Positional 2
USRTBL	User table	*Name*, ***NONE**	Optional, Positional 3
JOB	Job name	*Name*, **QFNCJOB**	Optional
JOBD	Job description	*Qualified object name*	Optional
	Qualifier 1: Job description	*Name*, **QFNC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MSGQ	Message queue	Single values: ***WRKSTN**, *NONE Other values: *Qualified object name*	Optional
	Qualifier 1: Message queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Device table (DEVTBL)

Specifies the name of the device table that the finance job uses to determine which 4704 or 3624 devices it controls. This is a required parameter.

Program table (PGMTBL)

Specifies the name of the program table that the finance job uses to determine, from the program ID (sent in the data stream with a finance transaction), which user program names will process the finance transaction. This is a required parameter.

User table (USRTBL)

Specifies the name of the user table that the finance job uses to verify a valid user when a finance sign-on is received.

The possible values are:

*NONE
    No user IDs are verified.

user-table-name
    Specify the name of a user table that defines user IDs for the 4700 device.

Job name (JOB)

Specifies the job name that is associated with the submitted finance job.

The possible values are:

QFNCJOB
    The job name is submitted as QFNCJOB.

job-name
    Specify the user-defined job name that is associated with the submitted finance job.

Job description (JOBD)

Specifies the name and library of the job description that is used by the finance job.

The possible values are:

*QFNC
    The submitted finance job uses the job description QFNC.

job-description-name
    Specify the name and library of a job description that is used by the finance job. (If no library name is given, the job description is found through the library list used by the job in which the Submit Finance Job (SBMFNCJOB) command is entered.

    The possible library values are:

    *LIBL

        The library list is used to locate the finance job.

    *CURLIB

The current library for the job is used to locate the finance job. If no library is specified as the library for the job, QGPL is used.

library-name

Specify the name of the library where the finance job is located.

Message queue (MSGQ)

Specifies the name of the message queue, to which messages can be sent while the finance job is running.

The possible values are:

WRKSTN
The finance messages are sent to the message queue of the work station from which the finance job was submitted.

*NONE
No finance messages are sent to a message queue.

message-queue-name
Specify the name and library of the user-defined message queue to which messages are sent. (If no library name is given, the library list of the job issuing the Submit Finance Job (SBMFNCJOB) command is used to find the queue.)

The possible library values are:

*LIBL   The library list is used to find the name of the message queue.

*CURLIB
The current library for the job is used to find the name of the message queue. If no library is specified as the current library for the job, QGPL is used.

library-name
Specify the name of the library where the name of the message queue is located.

Examples

Example 1: Submitting a Batch Job that Communicates with Devices it Acquires

```
SBMFNCJOB   DEVTBL(DEVTBL1)   PGMTBL(PGMTBL1)   USRTBL(USRTBL1)
```

This command submits batch job QFNCJOB. The job communicates with all devices it acquires from device table DEVTBL1, allowing users whose user IDs are found in USRTBL1 to sign on the devices. Each transaction sent by the finance devices is processed by determining, in PGMTBL1, which application program must be called, then it calls that program.

The job description used by the finance job in this example is QFNC. Messages sent as a result of the finance job are sent to the message queue of the work station from which the job was submitted.

Example 2: User IDs Not Verified

```
SBMFNCJOB   DEVTBL(DEVTBL2)   PGMTBL(PGMTBL2)
            JOB(CTFJOB)   JOBD(CTFJOBD)   MSGQ(*NONE)
```

This command submits batch job CTFJOB. CTFJOB runs under job description CTFJOBD and does not send messages to any work station message queue while running. No verification of user IDs is performed by the finance job.

Error messages

*ESCAPE Messages

CPF8382

Finance job cannot be processed.

Submit Job (SBMJOB)

The Submit Job (SBMJOB) command allows a job that is running to submit another job to a job queue to be run later as a batch job. Only one element of request data can be placed on the new job's message queue. The request data can be a CL command if the routing entry used for the job specifies a CL command processing program (such as the IBM-supplied QCMD program).

Note: A job started by the SBMJOB command uses the **Accounting code (ACGCDE)** of the job that submits the job. The ACGCDE specifications in the submitted job's job description (JOBD) and user's user profile (USRPRF) are ignored.

Restrictions:

1. The user that issues the SBMJOB command must have:
   - use (*USE) authority to the user profile specified on the **User (USER)** parameter.
   - use (*USE) authority to the command specified on the **Command to run (CMD)** parameter and execute (*EXECUTE) authority to the library containing that command.
   - use (*USE) authority to the job description specified on the **Job description (JOBD)** parameter and execute (*EXECUTE) authority to the library containing that job description.
   - use (*USE) authority to the job queue specified on the **Job queue (JOBQ)** parameter and execute (*EXECUTE) authority to the library containing that job queue.
   - use (*USE) and add (*ADD) authority to the message queue specified on the **Message queue (MSGQ)** parameter and execute (*EXECUTE) authority to the library containing that message queue.
   - job control (*JOBCTL) special authority to use the **Submitted for (SBMFOR)** parameter.
2. The user for the submitted job must have:
   - use (*USE) authority to the job description specified on the **Job description (JOBD)** parameter.
   - read (*READ) authority to the output queue specified on the **Output queue (OUTQ)** parameter and execute (*EXECUTE) authority to the library, in the submitted job's name space, that contains that output queue.
   - use (*USE) authority to all the libraries specified on the **System library list (SYSLIBL)** parameter.
   - use (*USE) authority to the library specified on the **Current library (CURLIB)** parameter.
   - use (*USE) authority to all the libraries specified on the **Initial library list (INLLIBL)** parameter.
   - use (*USE) authority to all auxiliary storage pool (ASP) device descriptions specified on the **Initial ASP group (INLASPGRP)** parameter.
   - use (*USE) authority to the sort sequence table specified on the **Sort sequence (SRTSEQ)** parameter and execute (*EXECUTE) authority to the library, in the submitted job's name space, that contains that sort sequence table.
3. This command is conditionally threadsafe. If a Job Notification Exit Point has been registered to send a message to a DDM data queue whenever a Submit Job is done, the message will not be sent if SBMJOB command is issued in a multithreaded job. For more information on the Job Notification function, refer to the Job Notification Exit Point in the Work Management chapter of the System API Reference manual.

Top

Parameters

Keyword	Description	Choices	Notes
CMD	Command to run	*Command string*	Optional
JOB	Job name	*Name*, ***JOBD**	Optional, Positional 1
JOBD	Job description	Single values: ***USRPRF** Other values: *Qualified object name*	Optional, Positional 2
	Qualifier 1: Job description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
JOBQ	Job queue	Single values: ***JOBD** Other values: *Qualified object name*	Optional, Positional 4
	Qualifier 1: Job queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
JOBPTY	Job priority (on JOBQ)	1-9, ***JOBD**	Optional
OUTPTY	Output priority (on OUTQ)	1-9, ***JOBD**	Optional
PRTDEV	Print device	*Name*, ***CURRENT**, *USRPRF, *SYSVAL, *JOBD	Optional
OUTQ	Output queue	Single values: ***CURRENT**, *USRPRF, *DEV, *JOBD Other values: *Qualified object name*	Optional
	Qualifier 1: Output queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
USER	User	*Name*, ***CURRENT**, *JOBD	Optional, Positional 3
PRTTXT	Print text	*Character value*, ***CURRENT**, *JOBD, *SYSVAL, *BLANK	Optional
RTGDTA	Routing data	*Character value*, **QCMDB**, *JOBD, *RQSDTA	Optional
RQSDTA	Request data or command	*Character value*, ***CMD**, *JOBD, *NONE, *RTGDTA	Optional
SYSLIBL	System library list	***CURRENT**, *SYSVAL	Optional
CURLIB	Current library	*Name*, ***CURRENT**, *USRPRF, *CRTDFT	Optional
INLLIBL	Initial library list	Single values: ***CURRENT**, *JOBD, *SYSVAL, *NONE Other values (up to 250 repetitions): *Name*	Optional
INLASPGRP	Initial ASP group	*Name*, ***CURRENT**, *JOBD, *NONE	Optional
LOG	Message logging	*Element list*	Optional
	Element 1: Level	0-4, ***JOBD**	
	Element 2: Severity	0-99, ***JOBD**	
	Element 3: Text	***JOBD**, *MSG, *SECLVL, *NOLIST	
LOGCLPGM	Log CL program commands	***JOBD**, *NO, *YES	Optional
LOGOUTPUT	Job log output	***JOBD**, *SYSVAL, *JOBLOGSVR, *JOBEND, *PND	Optional
JOBMSGQMX	Job message queue maximum size	2-64, ***JOBD**, *SYSVAL	Optional
JOBMSGQFL	Job message queue full action	***JOBD**, *SYSVAL, *NOWRAP, *WRAP, *PRTWRAP	Optional
INQMSGRPY	Inquiry message reply	***JOBD**, *RQD, *DFT, *SYSRPYL	Optional
HOLD	Hold on job queue	***JOBD**, *NO, *YES	Optional
SCDDATE	Schedule date	*Date*, ***CURRENT**, *MONTHSTR, *MONTHEND, *MON, *TUE, *WED, *THU, *FRI, *SAT, *SUN	Optional
SCDTIME	Schedule time	*Time*, ***CURRENT**	Optional
DATE	Job date	*Date*, ***JOBD**, *SYSVAL	Optional
SWS	Job switches	*Character value*, ***JOBD**	Optional
DSPSBMJOB	Allow display by WRKSBMJOB	***YES**, *NO	Optional

Keyword	Description	Choices	Notes
SBMFOR	Submitted for	Single values: ***CURRENT** Other values: *Qualified job name*	Optional
	Qualifier 1: Submitted for	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
MSGQ	Message queue	Single values: ***USRPRF**, *WRKSTN, *NONE Other values: *Qualified object name*	Optional
	Qualifier 1: Message queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRTSEQ	Sort sequence	Single values: ***CURRENT**, *USRPRF, *SYSVAL, *HEX, *LANGIDUNQ, *LANGIDSHR Other values: *Qualified object name*	Optional
	Qualifier 1: Sort sequence	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LANGID	Language ID	*Character value*, ***CURRENT**, *USRPRF, *SYSVAL	Optional
CNTRYID	Country or region ID	*Character value*, ***CURRENT**, *USRPRF, *SYSVAL	Optional
CCSID	Coded character set ID	1-65535, ***CURRENT**, *USRPRF, *SYSVAL, *HEX	Optional
CPYENVVAR	Copy environment variables	***NO**, *YES	Optional
ALWMLTTHD	Allow multiple threads	***JOBD**, *NO, *YES	Optional
SPLFACN	Spooled file action	***CURRENT**, *JOBD, *SYSVAL, *KEEP, *DETACH	Optional

<div align="right">Top</div>

Command to run (CMD)

Specifies a command that runs in the batch job, if the routing program used when this batch job is initiated is the IBM-supplied default routing program (QCMD). Because this command is used for the request data, this parameter is mutually exclusive with the **Request data or command (RQSDTA)** parameter.

command-string
> Specify the command that is run in the batch job. The command cannot be longer than 20000 characters.

<div align="right">Top</div>

Job name (JOB)

Specifies the name that is associated with the job when it is processed by the system.

*JOBD
> The simple name of the job description used with this job is the name of the job itself.

name   Specify the simple name of the job that is used while it is being processed by the system.

<div align="right">Top</div>

Job description (JOBD)

Specifies the job description used with the job.

Single values

*USRPRF
>The job description in the user profile under which the submitted job initially runs is used as the job description of the submitted job.

Qualifier 1: Job description

name   Specify the name of the job description used for the job.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
>The current library for the thread is used to locate the job description. If no library is specified as the current library for the thread, the QGPL library is used.

name   Specify the library where the job description is located.

<div align="right">Top</div>

Job queue (JOBQ)

Specifies the job queue in which this job is placed.

Single values

*JOBD
>The submitted job is placed on the job queue named in the specified job description.

Qualifier 1: Job queue

name   Specify the name of the job queue.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
>The current library for the thread is used to locate the job queue. If no library is specified as the current library for the thread, the QGPL library is used.

name   Specify the library where the job queue is located.

<div align="right">Top</div>

Job priority (on JOBQ) (JOBPTY)

Specifies the job queue scheduling priority. The highest priority is 1 and the lowest priority is 9.

*JOBD
>The scheduling priority specified in the job description is used for the job.

1-9   Specify the scheduling priority for the job.

<div align="right">Top</div>

Output priority (on OUTQ) (OUTPTY)

Specifies the output priority for spooled output files that are produced by this job. The highest priority is 1 and the lowest priority is 9.

*JOBD
> The output priority specified in the job description is used for the job.

1-9   Specify the priority of the submitted job's output files.

Print device (PRTDEV)

Specifies the qualified name of the default printer device for this job. If the printer file being used to create the output specifies to spool the file, the spooled file is placed on the device's output queue, which is named the same as the device.

Note: This assumes the defaults are specified on the **Output queue (OUTQ)** parameter of the printer file, job description, user profile and work station.

*CURRENT
> The same printer device being used by the job that is currently running is used by the submitted job.

*USRPRF
> The printer device specified in the user profile under which the submitted job initially runs is used as the printer device for this job. The printer device name is obtained from the profile when this command is run.

*SYSVAL
> The printer device specified in the system value, QPRTDEV, when this command is run is used for the submitted job.

*JOBD
> The printer device specified in the job description is used for the submitted job.

name   Specify the name of the printer device used for the submitted job.

Output queue (OUTQ)

Specifies the qualified name of the output queue used for spooled files that specify OUTQ(*JOB). This parameter applies only to printer files that have *JOB specified on the OUTQ parameter.

Single values

*CURRENT
> The output queue used by the job that is currently running is used for the submitted job.

*USRPRF
> The output queue in the user profile under which the submitted job initially runs is used as the output queue for the submitted job. The output queue name is obtained from the profile when this command is run.

*DEV   The output queue associated with the printer specified on the **Device (DEV)** parameter of the printer file is used. The output queue has the same name as the printer. The printer file DEV parameter is determined by the Create Printer File (CRTPRTF), Change Printer File (CHGPRTF), or the Override with Printer File (OVRPRTF) commands.

> **Note:** This assumes the defaults were specified on the OUTQ parameter for the printer file, job description, user profile, and workstation.

*JOBD

The output queue named in the job description used with the submitted job is the job's default output queue.

Qualifier 1: Output queue

name   Specify the name of the output queue that is used as the default output queue by the submitted job.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB

The current library for the thread is used to locate the output queue. If no library is specified as the current library of the thread, the QGPL library is used.

name   Specify the library where the output queue is located.

User (USER)

Specifies the name of the user profile for the job being submitted. If *RQD is specified in the job description, *JOBD cannot be specified. Instead, *CURRENT or a user-name must be specified.

Note: The following IBM-supplied objects are not valid for this parameter:
- QDBSHR
- QDFTOWN
- QDOC
- QLPAUTO
- QLPINSTALL
- QRJE
- QSECOFR
- QSPL
- QSYS
- QTSTRQS

*CURRENT

The same user profile used by the job that is currently running is used for the submitted job.

*JOBD

The user profile named in the specified job description is used for the job being submitted.

name   Specify the name of the user profile that is used for the job being submitted.

Print text (PRTTXT)

Specifies the text that is printed at the bottom of each page of printed output and on separator pages.

*CURRENT

The same print text of the submitting job is used.

*JOBD

The value in the job description is used for this job.

*SYSVAL

The system value, QPRTTXT, is used.

*BLANK

No text is printed.

character-value

Specify the character string that is printed at the bottom of each page. A maximum of 30 characters can be entered, enclosed in apostrophes.

Routing data (RTGDTA)

Specifies the routing data used to start the next routing step in the job. The routing data is used to determine the routing entry that identifies the program that the job runs.

QCMDB

The routing data used by the IBM-supplied subsystems to route batch jobs to the IBM-supplied control language processor QCMD is used.

*JOBD

The routing data specified in the job description is used.

*RQSDTA

The first 80 characters of the request data specified in the **Request data or command (RQSDTA)** parameter of this command are used as the routing data for the job.

character-value

Specify the character string that is used as the routing data for starting the job. A maximum of 80 characters can be entered, enclosed in apostrophes if necessary.

Request data or command (RQSDTA)

Specifies the request data that is placed as the last entry in this job's message queue. The request data can be a CL command to be run or a string of characters used by another program.

*CMD The input from the **Command to run (CMD)** parameter is placed in this job's message queue.

*JOBD

The request data specified in the job description used by the job is placed in this job's message queue.

*NONE

No request data is placed in the job's message queue.

*RTGDTA

The routing data specified in the **Routing data (RTGDTA)** parameter of this command is placed as the last entry in the job's message queue.

character-value

Specify the character string that is placed as the last entry in the submitted job's message queue. A maximum of 3000 characters can be entered, enclosed in apostrophes if necessary.

System library list (SYSLIBL)

Specifies the system portion of the initial library list that is used by the submitted job.

*CURRENT
> The same system library list being used by the thread that is currently running is used for the submitted job.

*SYSVAL
> The system default system library list is used by the job. It contains the library names that were specified in the system value, QSYSLIBL, at the time that the job is started.

Current library (CURLIB)

Specifies the name of the current library associated with the submitted job.

*CURRENT
> The current library being used by the thread that is currently running is used for the submitted job.

*USRPRF
> The current library in the user profile under which the submitted job initially runs is used as the current library for the submitted job.

*CRTDFT
> There is no current library for the submitted job. If objects are created in the current library, QGPL is used as the default current library.

name   Specify the library used as the current library of the submitted job.

Initial library list (INLLIBL)

Specifies the initial user part of the library list that is used to search for any object names that were specified without a library qualifier.

Note: Duplication of library names in the library list is not allowed.

Single values

*CURRENT
> The library list being used by the thread that is currently running is used for the submitted job.

*JOBD
> The library list in the job description used with this job is used as the initial user part of the library list for the job.

*SYSVAL
> The system default user library list is used by this job. It contains the library names that were specified in the system value QUSRLIBL at the time that the job is started.

*NONE
> The user portion of the initial library list is empty.

Other values (up to 250 repetitions)

name Specify the names of one or more libraries that are the user portion of the library list and are used by this job. The libraries are searched in the same order as they are listed.

Initial ASP group (INLASPGRP)

Specifies the initial setting for the auxiliary storage pool (ASP) group name for the initial thread of the submitted job. A thread can use the Set ASP Group (SETASPGRP) command to change its library name space. When an ASP group is associated with a thread, all libraries in the independent ASPs in the ASP group are accessible and objects in those libraries can be referenced using regular library-qualified object name syntax. The libraries in the independent ASPs in the specified ASP group plus the libraries in the system ASP (ASP number 1) and basic user ASPs (ASP numbers 2-32) form the library name space for the thread.

*CURRENT
The ASP group name for the current thread is used for the submitted job.

*JOBD
The initial ASP group name specified in the job description is used for the submitted job.

*NONE
Specifies the initial thread of the submitted job will be started with no ASP group. The library name space will not include libraries from any ASP group. Only the libraries in the system ASP and any basic user ASPs will be in the library name space.

name Specify the name of the ASP group to be set for the initial thread of the submitted job. The ASP group name is the name of the primary ASP device within the ASP group. All libraries from all ASPs in this ASP group will be included in the library name space.

Message logging (LOG)

Specifies the message logging values used to determine the amount and type of information sent to the job log by this job. This parameter has three elements: the message (or logging) level, the message severity, and the level of message text.

Element 1: Level

*JOBD
The value specified for message logging in the job description is used for this job.

0-4 Specify the message logging level used for this job's messages. The possible logging levels are:

0 No messages are logged.

1 All messages sent to the job's external message queue with a severity greater than or equal to the message logging severity are logged. This includes the indications of job start, job end, and job completion status.

2 The following information is logged:
- Logging level 1 information
- Request messages which result in a high-level message with a severity code greater than or equal to the message logging severity. Both the request message and all associated messages are logged.

> **Note:** A high-level message is one that is sent to the program message queue of the
> program that receives the request message. For example, QCMD is an IBM-supplied
> request processing program that receives request messages.

3   The following information is logged:
- Logging level 1 and 2 information
- All request messages
- Commands run by a CL program are logged if it is allowed by the logging of CL
  programs job attribute and the log attribute of the CL program.

4   The following information is logged:
- All request messages and all messages with a severity greater than or equal to the
  message logging severity, including trace messages.
- Commands run by a CL program are logged if it is allowed by the logging of CL
  programs job attribute and the log attribute of the CL program.

Element 2: Severity

*JOBD
       The value specified for message logging in the job description is used for this job.

0-99   Specify the message severity that is used in conjunction with the logging level to determine
       which error messages are logged in the job log.

Element 3: Text

*JOBD
       The value specified for message logging in the job description is used for this job.

*MSG   Only the message text is written to the job log.

*SECLVL
       Both the message text and the message help (cause and recovery) of the error message are written
       to the job log.

*NOLIST
       If the job ends normally, no job log is produced. If the job ends abnormally (if the job end code is
       20 or higher), a job log is produced. The messages that appear in the job log contain both the
       message text and the message help.

Log CL program commands (LOGCLPGM)

Specifies whether the commands that are run in a control language program are logged to the job log by
way of the CL program's message queue. This parameter sets the status of the job's logging flag. If *JOB
has been specified for the **Message logging (LOG)** parameter in the Create CL Program (CRTCLPGM)
command, the flag set in the **Log CL program commands (LOGCLPGM)** parameter is used. Other values
for the **Message logging (LOG)** parameter override the **Log CL program commands (LOGCLPGM)**
parameter. The commands are logged in the same manner as the requests.

*JOBD
       The value specified in the job description is used.

*NO   The commands in a CL program are not logged to the job log.

*YES   The commands in a CL program are logged to the job log.

Job log output (LOGOUTPUT)

Specifies how the job log will be produced when the job completes. This does not affect job logs produced when the message queue is full and the job message queue full action specifies *PRTWRAP. Messages in the job message queue are written to a spooled file, from which the job log can be printed, unless the Control Job Log Output (QMHCTLJL) API was used in the job to specify that the messages in the job log are to be written to a database file.

The job log output value can be changed at any time until the job log has been produced or removed. To change the job log output value for a job, use the Change Job (QWTCHGJB) API or the Change Job (CHGJOB) command.

The job log can be displayed at any time until the job log has been produced or removed. To display the job log, use the Display Job Log (DSPJOBLOG) command.

The job log can be removed when the job has completed and the job log has not yet been produced or removed. To remove the job log, use the Remove Pending Job Log (QWTRMVJL) API or the End Job (ENDJOB) command.

*JOBD
> The value specified in the job description is used.

*SYSVAL
> The value specified in the system value QLOGOUTPUT is used.

*JOBLOGSVR
> The job log will be produced by a job log server. For more information about job log servers, refer to the Start Job Log Server (STRLOGSVR) command.

*JOBEND
> The job log will be produced by the job itself. If the job cannot produce its own job log, the job log will be produced by a job log server. For example, a job does not produce its own job log when the system is processing a Power Down System (PWRDWNSYS) command.

*PND The job log will not be produced. The job log remains pending until removed.

Job message queue maximum size (JOBMSGQMX)

Specifies the maximum size of the job message queue.

*JOBD
> The value specified in the job description determines maximum size of the job message queue.

*SYSVAL
> The value in system value QJOBMSGQMX at the time the job is started is used as the maximum size of the job message queue.

2-64 Specify the maximum size, in megabytes, of the job message queue.

Job message queue full action (JOBMSGQFL)

Specifies the action that should be taken when the job message queue is full.

*JOBD

The value specified in the job description determines the action that should be taken.

*SYSVAL

The value specified for the QJOBMSGQFL system value is used.

*NOWRAP

The message queue does not wrap when it is full. This action ends the job.

*WRAP

The message queue wraps to the start of the message queue when full and starts filling the message queue again.

*PRTWRAP

The message queue wraps the job message queue when full and prints the messages that are being overlaid because of wrapping.

Inquiry message reply (INQMSGRPY)

Specifies the way that predefined messages that are sent as a result of running this job are answered. You can specify that the inquiry message reply control is taken from the job description, or that all inquiry messages require a reply, or that a default reply is issued, or that the system reply list is checked for a matching reply as each predefined inquiry message is sent.

*JOBD

The inquiry message reply control specified in the job description used with this job is used.

*RQD A reply is required by the receiver of the inquiry message for all inquiry messages that occur when this command is run.

*DFT The default message reply is used to answer any inquiry messages issued when this command is run.

*SYSRPYL

The system reply list is checked to see if there is an entry for any inquiry message that is issued as a result of running this job that has a message identifier and any comparison data that match the inquiry message identifier and message data. If a match occurs, the reply value in that entry is used. If no entry exists for that message, a reply is required.

Hold on job queue (HOLD)

Specifies whether this job is held at the time that it is put on the job queue. A job placed on the job queue in the hold state is held until it is released by the Release Job (RLSJOB) command or ended, either by the End Job (ENDJOB) command or by the Clear Job Queue (CLRJOBQ) command.

*JOBD

The value specified in the job description determines whether the job is held when it is put on the job queue.

*NO The job is not held when it is put on the job queue.

*YES The job is held when it is put on the job queue until it is released ended.

Schedule date (SCDDATE)

Specifies the date on which the scheduled job is submitted to the job queue and is eligible to run.

If your system or your job is configured to use the Julian date format, *MONTHSTR and *MONTHEND are calculated as if the system or job did not use the Julian date format.

*CURRENT

The submitted job becomes eligible to run on the current date.

*MONTHSTR

The submitted job becomes eligible to run on the first day of the month. If today is the first day of the month and the time you specify on the SCDTIME parameter has not passed, the job becomes eligible to run today. Otherwise, the job becomes eligible on the first day of the next month.

*MONTHEND

The submitted job becomes eligible to run on the last day of the month. If today is the last day of the month and the time you specify on the SCDTIME parameter has not passed, the job becomes eligible to run today. Otherwise, if today is the last day of the month and the time you specify on the SCDTIME parameter has passed, the job becomes eligible on the last day of the next month.

*MON

The job becomes eligible to run on Monday.

*TUE   The job becomes eligible to run on Tuesday.

*WED The job becomes eligible to run on Wednesday.

*THU The job becomes eligible to run on Thursday.

*FRI   The job becomes eligible to run on Friday.

*SAT   The job becomes eligible to run on Saturday.

*SUN   The job becomes eligible to run on Sunday.

date   Specify a date in the job date format with or without separators.

Schedule time (SCDTIME)

Specifies the time on the scheduled date at which the job becomes eligible to run.

Note: Although the time can be specified to the second, the load on the system may affect the exact time at which the submitted job becomes eligible to run.

The order that job entries with identical SCDDATE and SCDTIME values appear on the job queue may be different than the order in which they arrived. Likewise, these jobs may leave the job queue to be processed in an order different than the order in which they were entered. Do not assume jobs are entered or processed sequentially when they are scheduled to start at exactly the same time.

*CURRENT

The job is submitted at the current time.

time   Specify the time you want the job to start. The time is specified in 24-hour format and can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 to 23. Valid values for **mm** and **ss** range from 00 to 59.

- 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 this command is entered 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.

Job date (DATE)

Specifies the date that is assigned to the job when it is started.

*JOBD
> The date specified in the job description is the job date.

*SYSVAL
> The value in the QDATE system value at the time the job is started is the job date.

date   Specify the value that is used as the job date when the job is started; the date must be specified in the job date format.

Job switches (SWS)

Specifies the first settings for a group of eight job switches used with this job. These switches can be set or tested in a CL program and used to control the flow of the program. Only 0's (off) and 1's (on) can be specified in the 8-digit character string.

*JOBD
> The value specified in the job description is the first setting for the job's switches.

character-value
> Specify any combination of eight zeros and ones that is used as the first switch setting for the submitted job.

Allow display by WRKSBMJOB (DSPSBMJOB)

Specifies whether the job being submitted is allowed to be shown on the **Work with Submitted Jobs** panel. Any submitted job of the type specified by the **Jobs submitted from (SBMFROM)** parameter of the Work with Submitted Jobs (WRKSBMJOB) command can be shown if the job is not prevented by this parameter.

*YES   This job can be shown by the WRKSBMJOB command.

*NO   This job is not shown on any display produced by the WRKSBMJOB command.

Submitted for (SBMFOR)

Specifies the job name to be used on the **Jobs submitted from (SBMFROM)** parameter of the Work with Submitted Jobs (WRKSBMJOB) command.

Single values

Qualifier 1: Submitted for

name   Specify the name of the job.

Qualifier 2: User

name   Specify the user name that identifies the user profile under which the job is run.

Qualifier 3: Number

000000-999999
>   Specify the job number.

Note: You must have job control (*JOBCTL) special authority to use this parameter.

Message queue (MSGQ)

Specifies the message queue to which a completion message is sent when the submitted job has completed running, either normally or abnormally. If an abnormal ending occurs, the help information for the completion message specifies the possible causes.

Single values

*USRPRF
>   A completion message is sent to the message queue specified in the user profile of the submitter.

*WRKSTN
>   A completion message is sent to the work station message queue of the work station from which the job was submitted. If the job is submitted by a batch job, no completion message is sent.

*NONE
>   No completion message is sent.

Qualifier 1: Message queue

name   Specify the name of the message queue to which the completion message is sent.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
>   The current library for the thread is used to locate the message queue. If no library is specified as the current library for the thread, the QGPL library is used.

name   Specify the library where the message queue is located.

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons for this job.

Single values

*CURRENT

> The sort table specified for the job that is currently running is used.

*USRPRF

> The sort table specified in the user profile under which the submitted job initially runs is used. The user profile is specified on the **User (USER)** parameter.

*SYSVAL

> The system value QSRTSEQ is used.

*HEX   A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

*LANGIDUNQ

> A unique-weight sort table is used.

*LANGIDSHR

> A shared-weight sort table is used.

Qualifier 1: Sort sequence

name   Specify the name of the sort sequence table.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB

> The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is used.

name   Specify the library where the sort sequence table is located.

Language ID (LANGID)

Specifies the language identifier to be associated with this job. The language identifier is used when *LANGIDUNQ or *LANGIDSHR is specified on the **Sort sequence (SRTSEQ)** parameter. If the job CCSID is 65535, this parameter is also used to determine the value of the job default CCSID (DFTCCSID).

*CURRENT

> The language identifier specified for the job that is currently running is used.

*USRPRF

> The language ID specified in the user profile under which the submitted job runs is used. The user profile is specified on the **User (USER)** parameter.

*SYSVAL

> The system value QLANGID is used.

character-value

> Specify the language identifier to be used by the job.

Country or region ID (CNTRYID)

Specifies the country or region identifier (ID) to be used by the job.

*CURRENT

> The country or region ID specified for the job that is currently running is used.

*USRPRF

The country or region ID specified in the user profile under which the submitted job runs is used. The user profile is specified on the **User (USER)** parameter.

*SYSVAL

The system value QCNTRYID is used.

character-value

Specify the country or region ID to be used by the job.

Coded character set ID (CCSID)

Specifies the coded character set identifier (CCSID) used for the submitted job.

*CURRENT

The CCSID specified for the job that is currently running is used.

*USRPRF

The CCSID specified in the user profile where the submitted job initially runs is used.

*SYSVAL

The CCSID specified in the system value QCCSID at the time the job is started is used.

*HEX The CCSID 65535 is used.

1-65535

Specify the CCSID.

Copy environment variables (CPYENVVAR)

Specifies whether the environment variables from the submitting job are copied to the new job.

*NO The environment variables are not copied.

*YES The environment variables are copied.

Allow multiple threads (ALWMLTTHD)

Specifies whether or not the job is allowed to run with multiple user threads. This attribute does not prevent the operating system from creating system threads in the job. This attribute is not allowed to be changed after the job is submitted.

*JOBD

The value specified in the job description determines whether or not the job is allowed to run with multiple user threads.

*NO The job is not allowed to run with multiple user threads.

*YES The job is allowed to run with multiple user threads.

Spooled file action (SPLFACN)

Specifies whether or not spooled files are accessed through job interfaces after the job ends. Keeping spooled files with jobs allows job commands such as Work with Submitted Jobs (WRKSBMJOB) to work with the spooled files even after the job has ended. Detaching spooled files from jobs reduces the use of system resources by allowing job structures to be recycled when the jobs end.

<u>*CURRENT</u>
> The value from the current job is used for the submitted job.

*JOBD
> The value in the job description is used.

*SYSVAL
> The value specified in the system value QSPLFACN is used.

*KEEP
> When the job ends, if spooled files for the job exist in the system auxiliary storage pool (ASP 1) or in a basic user ASP (ASPs 2-32), the spooled files are kept with the job and the status of the job is updated to indicate that the job has completed. When all remaining spooled files for the job are in independent ASPs (ASPs 33-255), the spooled files will be detached from the job and the job will be removed from the system.

*DETACH
> When the job ends, the spooled files are detached from the job and the job is removed from the system.

Examples

Example 1: Submitting a Job

```
SBMJOB   JOB(SPECIAL)   JOBD(MYLIB/MYJOBD)   CMD(CALL MYPROG)
```

This command causes the job named SPECIAL to be submitted. Most of the attributes for the job are taken from the job description MYJOBD, or the job that is currently running, except for the command. The CALL command is placed on the submitted job's message queue so that the program MYPROG can be called and run later.

Example 2: Submitting a Job

```
SBMJOB   JOB(PAYROLL)   JOBD(PAYROLL)   INQMSGRPY(*RQD)
```

This command submits a job named PAYROLL to the system. All the information needed for this job (such as the job queue and routing data but not the inquiry message control value) is contained in the job description PAYROLL, or the job that is currently running. The library list in effect for the job issuing this command is used to find the job description. All inquiry messages sent during running of this job requires the receiver of the inquiry message to reply.

Example 3: Submitting a Job to a Job Queue

```
SBMJOB   JOBD(*USRPRF)   JOB(COPY12)   JOBQ(NIGHTQ)
         CMD(CPYF FILEA FILEB)
```

This command submits the job COPY12, which uses the job description in the user profile of the submitting job, to the job queue NIGHTQ. The CMD parameter provides the CL command necessary for the job to run. A command such as this might be used to copy the file at night while the system is unattended.

Error messages

*ESCAPE Messages

CPF133A

SBMJOB not allowed during IPL.

CPF1338

Errors occurred on SBMJOB command.

CPF1651

Sort sequence table not accessed.

Submit Network Job (SBMNETJOB)

Where allowed to run: All environments (*ALL)	
Threadsafe: No	

The Submit Network Job (SBMNETJOB) command sends an input stream to another user on the SNADS network. (The input stream is sent to another user where it can be filed, submitted, or rejected.) When the input stream arrives, its placement is governed by the job action (JOBACN) network attribute. If the value of JOBACN is *SEARCH, the entry in the network job table at the receiving system is used to determine the action taken. At the receiving system, the job may be submitted immediately, filed for placement by the receiving user, or rejected.

When the input stream arrives at the destination system, a message is sent to both the recipient of the input stream as well as the originator of the input stream stating that the input stream arrived. This command can only be used to send a batch input stream to a user on a remote system.

Restrictions:

1. To use this command, the user must have object operational and read authority to the file that is submitted, and for the library that contains the file.
2. The user must be enrolled in the system distribution directory to use this command. (For information on enrolling in the system distribution directory, see the SNA Distribution Services book, SC41-5410.
3. If the job action (JOBACN) network attribute on the receiving system is set to *SEARCH, there must be an entry for the user in the network job table on the receiving system. The entry in this table specifies a user profile on the receiving system that is used to verify that the user is authorized to submit the job on that system. The user profile on the receiving system must be authorized to use the job queues, and must have object operational authority for the job descriptions specified by the JOB commands in the input stream.
4. The file that is submitted cannot contain more than approximately 2 billion bytes of data.

Parameters

Keyword	Description	Choices	Notes
FILE	File	*Qualified object name*	Required, Positional 1
	Qualifier 1: File	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TOUSRID	User ID	Values (up to 50 repetitions): *Element list*	Required, Positional 2
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	
MBR	Member	*Name*, ***FIRST**	Optional, Positional 3
PTY	Send priority	***NORMAL**, *HIGH	Optional

File (FILE)

Specifies the name and library of the physical file containing the input stream that is sent.

This is a required parameter.

The possible library values are:

*LIBL   The library list is used to locate the file.

*CURLIB
>   The current library for the job is used to locate the file. If no library is specified as the current library for the job, QGPL is used.

library-name
>   Specify the library where the file is located.

User ID (TOUSRID)

Specifies the two-part user ID of one or more users to whom the input stream is submitted, or the name of one or more distribution lists containing the two-part user IDs of one or more users to whom the file is to be sent. A combination of both user IDs and distribution lists can be specified on the same command. Each user ID or distribution list is specified as a two-part name, and both parts are required.

NOTES:

1. Depending on the type of work station being used, the internal value for a user identifier may differ from the characters shown by the Display Directory Entries (DSPDIRE) command. If the byte-string value specified for the TOUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

2. The user specified in this parameter, or in the distribution list, must be a remote user. The SBMNETJOB command cannot be used to send input streams to local users.

You can enter multiple values for this parameter.

Member (MBR)

Specifies the member that is sent from the file.

*FIRST
>   The first member (in creation order) in the file is submitted.

member-name
>   Specify the name of the file member that is submitted.

Send priority (PTY)

Specifies the queuing priority used for the input stream when it is being routed through a SNADS network.

The possible values are:

*NORMAL

The input stream is sent with a service level priority of data low, which is used for most data traffic. On an AS/400 system, data low distributions are placed on the normal distribution queue specified for the route.

*HIGH

The input stream is sent with a service level priority of data high, which is used for high priority data traffic. On an AS/400 system, data high distributions are placed on the data high distribution queue specified for the route.

<div align="right">Top</div>

Examples

```
SBMNETJOB   FILE(PAYROLL)   TOUSRID(PAYROLL SYSTEM1)
            MBR(WEEKLY)
```

This command sends the input streams contained in member WEEKLY of file PAYROLL to user ID PAYROLL SYSTEM1.

<div align="right">Top</div>

Error messages

*ESCAPE Messages

CPF8056

File &1 in &2 not a physical file.

CPF8058

File &1 is a spooled file.

CPF8063

Cannot assign necessary resource.

CPF8065

Input stream &1 in &2 member &3 not sent to any users.

CPF8066

One or more user identifiers on this command is not correct.

CPF8068

Error detected while processing file to be sent.

CPF8072

Object to be sent is greater than maximum size of 2GB.

CPF9005

System resource required to complete this request not available.

CPF9006

User not enrolled in system distribution directory.

CPF9803

Cannot allocate object &2 in library &3.

CPF9807

One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

CPF9810

   Library &1 not found.

CPF9812

   File &1 in library &2 not found.

CPF9820

   Not authorized to use library &1.

CPF9822

   Not authorized to file &1 in library &2.

CPF9830

   Cannot assign library &1.

CPF9845

   Error occurred while opening file &1.

CPF9846

   Error while processing file &1 in library &2.

CPF9847

   Error occurred while closing file &1 in library &2.

CPF9848

   Cannot open file &1 in library &2 member &3.

CPF9849

   Error while processing file &1 in library &2 member &3.

Top

Submit Network Server Command (SBMNWSCMD)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Submit Network Server Command (SBMNWSCMD) command submits a command to run on the designated server. For SVRTYPE(*WINDOWS) or (*WINDOWSNT), command output is directed as specified by the CMDSTDOUT parameter. For all other types, command output is directed to the job log of the job that issues the command.

Parameters

Keyword	Description	Choices	Notes
CMD	Command	*Character value*, *NOLOGCMD	Required, Positional 1
SERVER	Server	*Character value*	Required, Positional 2
NOLOGCMD	Command (not logged)	*Character value*	Optional
SVRTYPE	Server type	***NWSUSRA**, *NWSA, *WINDOWS, *NETWARE, *WINDOWSNT	Optional
CMDTYPE	Command type	***SVRTYPE**, *WINDOWS, *NETWARE, *WINDOWSNT	Optional
LODUSRPRF	Load user profile	***YES**, *NO	Optional
AUTDMN	Authentication domain	*Character value*, ***PRIMARY**, *LOCAL	Optional
CMDSTDOUT	Command standard output	*Path name*, ***JOBLOG**, *PRINT	Optional
CVTSTDOUT	Convert standard output	***YES**, *NO	Optional

Command (CMD)

Specifies the command submitted to the network server. The syntax of the command string specified for this parameter is not checked.

*NOLOGCMD
Specifies the user wants to submit a command string that is not logged in the joblog. This is useful if the command string contains sensitive data such as passwords. If *NOLOGCMD is specified, a command string must be entered on the NOLOGCMD parameter.

command
The command string that is to be submitted to the network server. This command string will be shown in the joblog.

This is a required parameter.

Server (SERVER)

Specifies the name of the server to which the command is submitted.

This is a required parameter.

Command (not logged) (NOLOGCMD)

Specifies the command string to be submitted to the network server. This command string will not be echoed to the joblog. Use this parameter to submit commands that contain sensitive data such as passwords. This parameter is required if, and is only allowed when, CMD(*NOLOGCMD) is specified.

Note: Since the command is being executed on a server, the presence of this option does not prevent the server from returning the command string as part of the output data. Thus, depending on the command, it may still be returned and displayed in the job log or spool file.

Server type (SVRTYPE)

Specifies the type of server to which the command is sent.

*NWSUSRA
> The server type specified in the network server attributes (CHGNWSUSRA command) for the user profile running the SBMNWSCMD is used.

*NWSA
> The server type specified in the network server attributes (CHGNWSA command) is used.

*WINDOWS or *WINDOWSNT
> The server type is Windows. Only Windows commands will run on a local Windows server.
>
> **Note:** *WINDOWS should be used in V5R4 and later releases. The *WINDOWSNT value is supported for compatibility with releases prior to V5R4.

*NETWARE
> The server type is NetWare. Only NetWare commands can be sent to a NetWare server.

Command type (CMDTYPE)

Specifies the type of command string specified in the CMD parameter. This is used by the system to determine what type of processing needs to occur for the command string.

Note: This parameter is ignored.

*SVRTYPE
> The command string is processed based on the type of server specified in the SVRTYPE parameter. If the server is a NetWare server, the command string is assumed to be a NetWare command. If the server is a *WINDOWS or a *WINDOWSNT server, the command string is assumed to be a Windows command.

*WINDOWS or *WINDOWSNT
> The command string is a Windows command.

Note: *WINDOWS should be used in V5R4 and later releases. The *WINDOWSNT value is supported for compatibility with releases prior to V5R4.

*NETWARE
> The command string is a NetWare command.

Load user profile (LODUSRPRF)

Specifies whether or not to load the user profile on the remote server as part of the login environment.

Note: This parameter is only valid when SVRTYPE(*WINDOWS) or (*WINDOWSNT) is specified.

*YES   Load the user profile.

*NO   Do not load the user profile.

Authentication domain (AUTDMN)

Specifies the Windows domain where the user is authenticated.

Note: This parameter is only valid when SVRTYPE(*WINDOWS) or (*WINDOWSNT) is specified.

*PRIMARY
> The user is authenticated on the primary domain of the server.

*LOCAL
> The user is authenticated on the local server.

'domain-name'
> Specify the domain name where the user is authenticated.

Command standard output (CMDSTDOUT)

Specifies where the standard output returned from the command is to be stored, if any exists. Standard output can be written to the job log of the job that issues this command, it can be written to a spooled file, or it can be written to a file. The standard error returned from the command is always directed to the job log of the job that issues this command, if any exists.

Note: This parameter is only valid when SVRTYPE(*WINDOWS) or (*WINDOWSNT) is specified.

*JOBLOG
> The standard output of the network server command is directed to the job log of the job that issues this command. It shares the job log with the standard error output of the network server command. Both may be mixed in the job log, depending on the order by which the command writes standard output and standard error information.

*PRINT
> The standard output of the network server command is directed to a spooled file. If CVTSTDOUT(*YES) is specified, certain control characters such as line feeds and carriage returns are converted to new lines and other non-displayable control characters such as highlight and underscore are converted to blanks.

'stream-file-path-name'
>   Specify the path name of the stream file to which the standard output of the network server command is directed. The specified path must exist. If the stream file doesn't exist, it is created. If the stream file exists, all data is overlaid. Additional information about path name is in the Integrated file system information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter book.

<div align="right">Top</div>

Convert standard output (CVTSTDOUT)

Specifies whether the standard output will be converted from the server's code set to the CCSID of the i5/OS user profile that submitted the command. For binary output, CVTSTDOUT(*NO) should be specified.

Note: This parameter is only valid when CMDSTDOUT(*PRINT) or CMDSTDOUT(stream-file-path-name) is specified.

*YES   The output is converted from the server's code set to the CCSID of the i5/OS user profile that submitted the command.

*NO   The output is not converted from the server's code set.

<div align="right">Top</div>

Examples

Example 1: Submitting a Windows Command

```
SBMNWSCMD   CMD('net config server')  SERVER(WINSVR)
            CMDSTDOUT(*JOBLOG)
```

This command will display the Windows Server service settings on the Windows server WINSVR. Standard output from the command is returned to i5/OS and directed to the job log.

Example 2: Submitting a NetWare command

```
SBMNWSCMD   CMD('CONFIG')  SERVER(NTW01)  SVRTYPE(*NETWARE)
```

This command submits the NetWare CONFIG command to run on the server named NTW01. Output is returned to the job log.

<div align="right">Top</div>

Error messages

ESCAPE Messages

CPFA43F
>   Network server command not submitted.

CPFA46C
>   Unable to complete command processing on server &1.

CPFA46F
>   Network server description &1 not found.

<div align="right">Top</div>

Submit Remote Command (SBMRMTCMD)

The Submit Remote Command (SBMRMTCMD) command sends a CL command through Distributed Data Management (DDM) to be run on the target system specified by a DDM file. The DDM file determines the communications line used, and indirectly identifies the target system that receives the submitted command.

This command sends only CL commands to a remote system which supports the SBMRMTCMD command language. It cannot send non-CL commands to a different system type (for example, OCL commands cannot be sent to a target System/36).

The primary purpose of this command is to allow a source system user or program to perform file management operations and file authorization activities on files located on a target system.

- Create or delete physical, logical, device, or source files
- Grant or revoke object authority to remote files
- Check, rename, or move files or other objects
- Save or restore files or other objects

Restrictions:

- File processing on the target system operates independently of processing on the source system. Commands dependent on a specific recursion level or request level may *not* function as expected.
- Output (such as spooled files) generated by a submitted command exists only on the target system. The output is *not* sent back to the source system. Therefore display commands or commands that are used to service programs should not be sent because the output results remain at the target system.
- Translation is not performed for *impromptu* messages caused by target system errors, because they are not stored on the system; the text for an impromptu message is sent directly to the source system and displayed. The message identifier of all other message types generated on the remote system is sent back to the source system. The message text that exists for the message identifier on the source system is displayed as it would be for the same error on the source system.
- A maximum of 10 messages, generated during the running of a submitted command, can be sent by the target system to the source system. If more than 10 messages are generated, an additional *informational* message is sent that indicates that the messages exist in the job log for the target job on the target system. If one of those messages is an *escape* message, the first nine messages of other types are sent, followed by the informational message and the escape message.
- In multithreaded jobs, this command is not threadsafe and fails for DDM files of type *SNA.

Top

Parameters

Keyword	Description	Choices	Notes
CMD	Command to run	*Character value*	Required, Positional 1

Keyword	Description	Choices	Notes
DDMFILE	DDM file	*Qualified object name*	Required, Positional 2
	Qualifier 1: DDM file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

<div align="right">Top</div>

Command to run (CMD)

Specifies a character string of up to 2000 characters that represents a command that is run on the target system. The command must be allowed in both batch and interactive environments on the target system. The command must be enclosed in apostrophes if it contains embedded blanks or special characters. The command must use the syntax of the target system.

This is a required parameter.

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. This is required to allow coding a quoted string within another quoted string.

<div align="right">Top</div>

DDM file (DDMFILE)

Specifies the name and library of the Distributed Data Management (DDM) file that is used to submit the command to the target system. The DDM file is used only to determine the remote location representing the target system. The remote file name associated with the DDM file is ignored by this command.

This is a required parameter.

The possible library values are:

*LIBL   The library list is used to locate the file.

*CURLIB
> The current library for the job is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

library-name
> Specify the library where the file is located.

<div align="right">Top</div>

Examples

Example 1: Deleting a File

```
SBMRMTCMD   CMD('DLTF LIBX/FRED')  DDMFILE(DENVER)
```

This command deletes the file named FRED in library LIBX on the target system that is associated with the DDM file named DENVER.

Example 2: Creating a Physical File

```
SBMRMTCMD   CMD('CRTPF  SALES/CAR  QGPL/QDDSSRC MASTER')
            DDMFILE(DENVER)
```

This command creates the physical file CAR in library SALES using the data description specifications (DDS) in the source file QDDSSRC and source member named MASTER in the QGPL library. The DDS must already exist on the target system identified by the DDM file named DENVER in the target job's library list.

Example 3: Changing the Text Description

```
SBMRMTCMD   CMD('CHGDDMF  FILE(LIBX/STANLEY)
              TEXT(''Don''''t forget to pair apostrophes.'')')
            DDMFILE(SMITH)
```

This command changes the text in the description of the DDM file named STANLEY which is stored in library LIBX. Because the submitted command requires an outside set of single apostrophes (for the CMD parameter), each single or double apostrophe normally required in the TEXT parameter for *local* system processing must be doubled for *remote* system processing. The coding above produces a single apostrophe in the text when it is shown or printed on the remote system.

Example 4: Creating a DDM File

```
SBMRMTCMD   CMD('CRTDDMF FILE(SALES/MONTHLY)
                    RMTFILE(*NONSTD ''CAR.SALES(JULY)''
                    RMTLOCNAME(DALLAS)')
            DDMFILE(CHICAGO)
```

This command creates (on the target system identified by the information in the DDM file named CHICAGO) another DDM file named MONTHLY. The new DDM file is stored in a library named SALES on the CHICAGO system. The new DDM file on the CHICAGO system is used to access a file and *member* on a different system named DALLAS. The accessed file is named SALES/CAR and the member name in the file is JULY.

Note that this CRTDDMF command string contains *three* sets of single apostrophes: one set to enclose the entire command being submitted, and a double set to enclose the file and member named in the RMTFILE parameter. This is how any i5/OS file *member* name must be specified on the SBMRMTCMD command, because of the parentheses needed to enclose the member name.

Example 5: Replacing a Portion of the Library List

```
SBMRMTCMD   CMD('RPLLIBL  LIBL(QGPL QTEMP SALES EVANS)')
            DDMFILE(EVANS)
```

This command replaces the user's portion of the library list being used by the target job associated with the DDM file named EVANS, which is being used by the source job in which this SBMRMTCMD command is being submitted. In that source job, if there are other open DDM files that specify the same device and mode, this library list is used for them also.

Top

Error messages

*ESCAPE Messages

CPF9164
> Target system does not support SBMRMTCMD.

CPF9165
> File &1 in library &2 not a DDM File.

CPF917A
> Error occurred on distributed file.

CPF917B

      Target system &3 not available.

CPF9172

      SBMRMTCMD command ended abnormally.

CPF9174

      Error on call to user exit program on target system.

CPF9175

      Error during processing of user exit program.

CPF9177

      User exit program did not complete successfully.

CPF9178

      Processing of the command specified by SBMRMTCMD failed.

CPF9182

      Cannot start DDM communications.

Top

Select (SELECT)

Where allowed to run:
- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: Yes

The Select (SELECT) command begins a control structure to allow conditional processing of command groups. The select group is ended with an End Select (ENDSELECT) command and must contain one or more When (WHEN) commands and optionally an otherwise (OTHERWISE) command.

When a Select command is entered, the condition specified on the first When command is evaluated. If the condition is true, the command specified on the THEN parameter is run. After the command or group of commands is processed, control will pass to the matching ENDSELECT command. Only one WHEN or the optional OTHERWISE will be processed within a SELECT group.

If the first WHEN condition is not true, subsequent WHEN commands are evaluated in the order specified. If no WHEN command condition is true, control will pass to the command specified on the OTHERWISE command, if OTHERWISE is specified.

Restrictions:
- This command is valid only within a CL procedure.
- Up to 25 levels of nested SELECT commands are allowed.

There are no parameters for this command.

Parameters

None

Examples

```
DCL   VAR(&INT)  TYPE(*INT)  LEN(4)
 :
SELECT
  WHEN   COND(&INT *LT 0)  THEN(DO)
    :   (group of CL commands)
  ENDDO
  WHEN   COND(&INT *EQ 0)    /* Do nothing when zero */
  WHEN   COND(&INT *GT 0)  THEN(CHGVAR &INT (&INT - 1)
ENDSELECT
```

The SELECT command group will evaluates the conditions of the WHEN commands in the order they are encountered. When a COND parameter results in a true value, the command specified on the THEN parameter is processed. After the command on the THEN parameter is processed, control passes to the command following the ENDSELECT command.

Error messages

None

Set ASP Group (SETASPGRP)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Set Auxiliary Storage Pool Group (SETASPGRP) command sets the auxiliary storage pool (ASP) group for the current thread. Additionally, this command allows you to change the libraries in the library list for the current thread. If an ASP group had already been set, this command will remove the old ASP group from the current thread and set the specified ASP group for the current thread. Once the specified ASP group has been set for the current thread, all libraries in the independent ASPs in the ASP group are accessible and objects in those libraries can be referenced using regular library-qualified object name syntax. The libraries in the independent ASPs in the specified ASP group plus the libraries in the system ASP (ASP number 1) and basic user ASPs (ASP numbers 2-32) form the *library name space* for the thread. All libraries in the library list need to be in the new library name space or the library list is not changed and the new ASP group is not set.

Restrictions:

- You must have use (*USE) authority to all ASP device descriptions in the ASP group and to all the specified libraries in the library list before the library name space and the library list are changed. If you are not authorized to an ASP device description or to one of the libraries, the ASP group will not be set and the library list will not be changed.

- When *CURUSR is specified for the **ASP group (ASPGRP)** or **Libraries for current thread (USRLIBL)** parameter, you must have read (*READ) authority to the job description listed in your user profile and execute (*EXECUTE) authority to the library where the job description is located.

- The SETASPGRP command is not allowed in the following:

  - System jobs QPFRADJ, QJOBSCD, QSYSARB, QSYSARB3, QSYSARB4 and QLUS.

  - All subsystem monitor jobs.

  - DDM, DRDA, database host server and SQL server jobs once the initial namespace has been established.

  - Receive Journal Entry (RCVJRNE) and Delete Journal Receiver (DLTJRNRCV) command exit programs.

  - Management Central Registered Inventory Gathering Service (RIGS) Exit Programs (exit point QIBM_QYIV_INVGTRSRV).

  - Trigger or format selector programs that run as part of database I/O operations.

  - Attention programs (the PGM parameter of the SETATNPGM command).

  - Break handling programs (the PGM parameter of the CHGMSGQ command).

  - Programming Development Manager (PDM) functions.

Parameters

Keyword	Description	Choices	Notes
ASPGRP	ASP group	*Name*, *CURUSR, *NONE	Required, Positional 1
SYSLIBL	System library list	***CURSYSBAS**, *SYSVAL	Optional
CURLIB	Current library	*Name*, ***CURSYSBAS**, *CURUSR, *CRTDFT	Optional

Keyword	Description	Choices	Notes
USRLIBL	Libraries for current thread	Single values: ***CURSYSBAS**, *CURUSR, *SYSVAL, *NONE Other values (up to 250 repetitions): *Name*	Optional

ASP group (ASPGRP)

Specifies the name of the auxiliary storage pool (ASP) group to set for the current thread. The ASP group name is the name of the *primary* ASP device within the ASP group.

This is a required parameter.

*CURUSR
> The ASP group is set to the value defined for the **Initial ASP group (INLASPGRP)** in the default job description of the user profile that the thread is currently running under.

*NONE
> Specify for the current thread to have no ASP group. The library name space will not include libraries from any ASP group. Only the libraries in the system ASP and any basic user ASPs will be in the library name space.

name Specify the name of the *primary* ASP in the ASP group to be set for the current thread. All libraries from all ASPs in this ASP group will be included in the library name space.

System library list (SYSLIBL)

Specifies the system part of the library list for the thread in which the command is entered.

*CURSYSBAS
> The libraries in the system part of the library list of the current thread that are found in the system ASP (ASP number 1) or any configured basic user ASP (ASP numbers 2-32) are used as the new system part of the library list.

*SYSVAL
> The system part of the library list of the current thread is set from the current value of system value QSYSLIBL.

Current library (CURLIB)

Specifies the library to be used in the current library entry of the library list for the thread. If *CURUSR or a library name is specified and the library cannot be found in the new library name space, an error message is sent and the library list and ASP group are not changed.

*CURSYSBAS
> The library name in the current library entry of the library list is used as the new current library if the library is found in the system ASP (ASP number 1) or any configured basic user ASP (ASP numbers 2-32). If the library name in the current entry is not found in the system ASP or any basic user ASP, the current library entry will be removed from the library list.

*CURUSR

The current library is set to the value defined for the **Current library (CURLIB)** attribute of the user profile that the thread is currently running under.

*CRTDFT

Changes the library list to remove any name from the current library entry. If objects are created specifying *CURLIB for the library name, library QGPL will be used.

name   Specify the name of the library that replaces the current library entry in the library list.

Libraries for current thread (USRLIBL)

Specifies the libraries that are placed in the user part of the library list. If *CURUSR or a list of library names is specified and any of these libraries cannot be found in the new library name space, an error message is sent and the library list and ASP group are not changed.

Single values

*CURSYSBAS

The libraries in the user part of the library list of the current thread that are found in the system ASP (ASP number 1) or any configured basic user ASP (ASP numbers 2-32) are used as the new user part of the library list.

*CURUSR

The user part of the library list for the thread is set to the value defined for the **Initial library list (INLLIBL)** in the default job description of the user profile that the thread is currently running under.

*SYSVAL

The user part of the library list is set from the current value of system value QUSRLIBL.

*NONE

Changes the user part of the library list to remove all library names.

Other values (up to 250 repetitions)

name   Specify the names of the libraries to be used as the user part of the library list, in the order in which they are to be searched.

Examples

Example 1: Set New ASP Group

```
SETASPGRP  ASPGRP(WAREHOUSE1) SYSLIBL(*SYSVAL)  CURLIB(*CURUSR)
           USRLIBL(*CURSYSBAS)
```

This command will set the auxiliary storage pool (ASP) group for the thread in which the command runs to be WAREHOUSE1. This will change the library name space for the thread to include all libraries in any of the independent ASPs in the ASP group identified by the independent ASP device named WAREHOUSE1. The system part of the library list will be set from system value QSYSLIBL. The current library entry of the library list will be set from the *Current library* value defined in the user profile that the thread is currently running under. The user part of the library list will be set using the current user part of the library list and removing any libraries that are not found in the system ASP or configured basic user ASPs.

Example 2: Set to No ASP Group

```
SETASPGRP  ASPGRP(*NONE)  SYSLIBL(*CURSYSBAS)  CURLIB(*CRTDFT)
           USRLIBL(*NONE)
```

This command will remove any ASP group for the thread in which the command runs. This will change the library name space for the thread to include only those libraries in the system ASP (ASP number 1) and basic user ASPs (ASP numbers 2-32). The system part of the library list will be set using the current system part of the library list and removing any libraries that are not found in the system ASP or configured basic user ASPs. The current library entry of the library list will be changed to be empty which will cause library QGPL to be used as the current library. The user part of the library list will be changed to be empty.

Top

Error messages

*ESCAPE Messages

CPFB8E9

ASP group &1 not set for thread &2.

Top

Set Attention Program (SETATNPGM)

Where allowed to run: Interactive environments (*INTERACT
*IPGM *IREXX *EXEC)
Threadsafe: No

Parameters
Examples
Error messages

The Set Attention Program (SETATNPGM) command sets up a program that is called when the Attention key is pressed. The setting is in effect for this recursion level and lower levels if more programs are called, but it is no longer in effect if the job returns from this recursion level to the previous one. If the Attention key handler's status is on, the specified program is called when the key is pressed. No parameters are passed to the Attention key handler when it is called. The Attention handling program runs in the same process with the same job attributes, overrides, and group authorities as the program that issued the SETATNPGM command. However, program adopted authority is not carried over. The Attention handling program must not reside in an independent auxiliary storage pool (ASP). It must reside in the system ASP or a basic user ASP.

Top

Parameters

Keyword	Description	Choices	Notes
PGM	Program	Single values: *CURRENT, *PRVINVLVL Other values: *Qualified object name*	Required, Positional 1
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SET	Set attention key	***ON**, *OFF	Optional, Positional 2

Top

Program (PGM)

Specifies the qualified name of the program to be the Attention key handler at this recursion level.

This is a required parameter.

Single values

*CURRENT
> The program name of the Attention key handler currently in effect is used as the value of this parameter.

*PRVINVLVL
> The Attention key handler in effect at the previous recursion level is reinstated as the Attention key handler at this recursion level. The **Set attention key (SET)** parameter is not allowed if this special value is specified, because the SET status of the previous recursion level is also reinstated. This option is used when a program has specified an Attention program and wants to revert back to a previous level.

Qualifier 1: Program

name   Specify the name of the Attention key handler program.

Qualifier 2: Library

*LIBL   All libraries in the thread's library list are searched until a match is found.

*CURLIB
> The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name   Specify the library where the attention program is located.

Set attention key (SET)

Specifies whether the Attention key handler indicated in the **Program (PGM)** parameter is called when the Attention key is pressed. This parameter is not allowed if *PRVINVLVL is specified for the PGM parameter.

*ON   The Attention key handler specified in the **Program** (PGM) parameter is called when the Attention key is pressed.

*OFF   The Attention key handler specified in the **Program** (PGM) parameter is not called when the Attention key is pressed.

Examples

Example 1: Setting the ATTN Key Handler

```
SETATNPGM    PGM(QGPL/ATTN)   SET(*ON)
```

This command causes the program QGPL/ATTN to become the ATTN key handler. Because SET(*ON) is specified, the program is called when the ATTN key is pressed.

Example 2: Setting the Attention Key Off

```
SETATNPGM    PGM(*CURRENT)   SET(*OFF)
```

The current attention handling program has its status changed to SET(*OFF). Because the status is SET(*OFF) when the ATTN key is pressed, the attention handling program is not called.

Example 3: Previous Recursion-Level Support

```
SETATNPGM    PGM(*PRVINVLVL)
```

The attention handling program and status that was in effect at the previous recursion level is reinstated at this recursion level. If no attention handler is in effect, after this command is run nothing happens when the ATTN key is pressed.

Example 4: Emulating the System Request Key

```
SETATNPGM    PGM(QWSSYSRQ)
```

The system-supplied program QWSSYSRQ will be called when the ATTN key is pressed. This system program allows the ATTN key to act as a system request key by showing the system request menu on the display when the ATTN key is pressed.

Error messages

CPF1318

     Attention key program &1 in &2 not set.

Top

Set Customization Data (SETCSTDTA)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Set Customization Data (SETCSTDTA) command can be used to copy and delete information in order to customize data, such as work area content and mouse double-click action values, for Graphical Operations support.

Using this command, administrators can set up users by getting information from a named user profile and copying it to other user profiles. Administrators can also remove the customized values for a named user, and choose to replace those customized values with data from a named user profile.

Note: 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.

Error messages for SETCSTDTA

*ESCAPE Messages

CPF2204
　　　　User profile &1 not found.

CPF2217
　　　　Not authorized to user profile &1.

GUI0085
　　　　&1 does not have customization data.

GUI0087
　　　　Some user profiles did not have customization data copied successfully.

GUI0091
　　　　Some user profiles did not have customization data deleted successfully.

GUI0117
　　　　&1 is not valid for subset key.

Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	Values (up to 300 repetitions): *Simple name*	Required, Positional 1
VALUE	Customization data to use	*Simple name*, *NONE	Required, Positional 2
REPLACE	Replace customization data	<u>*NO</u>, *YES	Optional, Positional 3

User profile (USRPRF)

Specifies the names of the user profiles for which customized values are to be set or removed with the option to set them again. A maximum of 300 user profiles can be named.

You can enter multiple values for this parameter. If you are on an entry display and you need additional entry fields to enter these multiple values, type a plus sign (+) in the entry field opposite the phrase "+ for more", and press the Enter key.

This is a required parameter.

Customization data to use (VALUE)

Specifies the customization data value to which the user profile named on the **User profile** prompt (USRPRF parameter) is to be set.

The possible values are:

*NONE
   The customization data is to be removed from the user profile and no customization value is to be set.

user-profile
   Specify the name of the user profile from which the customization value data is to be copied and set.

This is a required parameter.

Replace customization data (REPLACE)

Specifies whether to replace existing customization data if the user profile named on the **User profile** prompt (USRPRF parameter) has customization data.

The possible values are:

*NO   The existing customization data is not replaced. An exception message is issued.

*YES   The existing customization data is replaced with the customization data from the user profile specified on the **Customization data to use** prompt (VALUE parameter).

Examples

None

Error messages

*ESCAPE Messages

CPF2204

      User profile &1 not found.

CPF2217

      Not authorized to user profile &1.

GUI0085

      &1 does not have customization data.

GUI0087

      Some user profiles did not have customization data copied successfully.

GUI0091

      Some user profiles did not have customization data deleted successfully.

GUI0117

      &1 is not valid for subset key.

Top

Set Keyboard Map (SETKBDMAP)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Set Keyboard Map (SETKBDMAP) command allows the user to override the PA (Program Attention) and PF (Program Function) key assignment defaults provided by the system.

This command assigns the specified F-to-PF map to the device where the command was entered (if it is a 3270 display station device) or to the 3270 display station specified if the user has authority to that device. More information on the user-assignment keyboard mapping is in Remote Work Station Support book, SC41-5402 book.

Parameters

Keyword	Description	Choices	Notes
DEV	Device	*Name*, ***REQUESTER**	Optional, Positional 1
PF1	PF1 key value	***HELP**, *ATTN, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF2	PF2 key value	*ATTN, *HELP, ***HLP3270**, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF3	PF3 key value	*ATTN, *HELP, *HLP3270, ***CLEAR**, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF4	PF4 key value	*ATTN, *HELP, *HLP3270, *CLEAR, ***PRINT**, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF5	PF5 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, ***DSPATR**, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF6	PF6 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, ***TEST**, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF7	PF7 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, ***DOWN**, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional

Keyword	Description	Choices	Notes
PF8	PF8 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, ***UP**, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF9	PF9 key value	*NONE, ***ATTN**, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF10	PF10 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, ***RESET**, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF11	PF11 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, ***SYSREQ**, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF12	PF12 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, ***BCKSPC**, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF13	PF13 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, ***F13**, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF14	PF14 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, ***F14**, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF15	PF15 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, ***F15**, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF16	PF16 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, ***F16**, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF17	PF17 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, ***F17**, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF18	PF18 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, ***F18**, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF19	PF19 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, ***F19**, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional

Keyword	Description	Choices	Notes
PF20	PF20 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, **<u>*F20</u>**, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF21	PF21 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, **<u>*F21</u>**, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF22	PF22 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, **<u>*F22</u>**, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF23	PF23 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, **<u>*F23</u>**, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PF24	PF24 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, **<u>*F24</u>**, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA1PF1	PA1-PF1 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, **<u>*F1</u>**, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA1PF2	PA1-PF2 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, **<u>*F2</u>**, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA1PF3	PA1-PF3 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, **<u>*F3</u>**, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA1PF4	PA1-PF4 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, **<u>*F4</u>**, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA1PF5	PA1-PF5 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, **<u>*F5</u>**, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA1PF6	PA1-PF6 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, **<u>*F6</u>**, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA1PF7	PA1-PF7 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, **<u>*F7</u>**, *F8, *F9, *F10, *F11, *F12	Optional

Keyword	Description	Choices	Notes
PA1PF8	PA1-PF8 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, **_*F8_**, *F9, *F10, *F11, *F12	Optional
PA1PF9	PA1-PF9 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, **_*F9_**, *F10, *F11, *F12	Optional
PA1PF10	PA1-PF10 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, **_*F10_**, *F11, *F12	Optional
PA1PF11	PA1-PF11 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, **_*F11_**, *F12	Optional
PA1PF12	PA1-PF12 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, **_*F12_**	Optional
PA2PF1	PA2-PF1 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, **_*F13_**, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA2PF2	PA2-PF2 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, **_*F14_**, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA2PF3	PA2-PF3 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, **_*F15_**, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA2PF4	PA2-PF4 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, **_*F16_**, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA2PF5	PA2-PF5 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, **_*F17_**, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA2PF6	PA2-PF6 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, **_*F18_**, *F19, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA2PF7	PA2-PF7 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, **_*F19_**, *F20, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional

Keyword	Description	Choices	Notes
PA2PF8	PA2-PF8 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, **_F20_**, *F21, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA2PF9	PA2-PF9 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, **_F21_**, *F22, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA2PF10	PA2-PF10 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, **_F22_**, *F23, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA2PF11	PA2-PF11 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, **_F23_**, *F24, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional
PA2PF12	PA2-PF12 key value	*ATTN, *HELP, *HLP3270, *CLEAR, *PRINT, *DSPATR, *TEST, *DOWN, *UP, *NONE, *RESET, *SYSREQ, *BCKSPC, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, **_F24_**, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12	Optional

Device (DEV)

Specifies a valid 3270 display station that is assigned this keyboard mapping function.

The possible values are:

*REQUESTER
> This mapping is assigned to the device where the command is entered.

device-name
> Specify the name of the device to which the new keyboard mapping is to apply. You must have allocation rights to the specified device.

PF1 key value (PF1)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP   Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF2 key value (PF2)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
5250 Help

*HLP3270
3270 Help Text (Display Active Keyboard Map)

*CLEAR
Clear Screen

*PRINT
Print Screen

*DSPATR
Display Imbedded Attributes

*TEST Test Request

*DOWN
Roll Down

*UP Roll Up

*NONE
No Assignment

*RESET
Error Reset

*SYSREQ
System Request

*BCKSPC
Record Backspace

*ATTN
Attention

*F1-*F24
F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF3 key value (PF3)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
    5250 Help

*HLP3270
    3270 Help Text (Display Active Keyboard Map)

*CLEAR
    Clear Screen

*PRINT
    Print Screen

*DSPATR
    Display Imbedded Attributes

*TEST Test Request

*DOWN
    Roll Down

*UP   Roll Up

*NONE
    No Assignment

*RESET
    Error Reset

*SYSREQ
    System Request

*BCKSPC
    Record Backspace

*ATTN
    Attention

*F1-*F24
    F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF4 key value (PF4)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
5250 Help

*HLP3270
3270 Help Text (Display Active Keyboard Map)

*CLEAR
Clear Screen

*PRINT
Print Screen

*DSPATR
Display Imbedded Attributes

*TEST Test Request

*DOWN
Roll Down

*UP Roll Up

*NONE
No Assignment

*RESET
Error Reset

*SYSREQ
System Request

*BCKSPC
Record Backspace

*ATTN
Attention

*F1-*F24
F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF5 key value (PF5)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

Top

PF6 key value (PF6)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET

Error Reset

*SYSREQ

System Request

*BCKSPC

Record Backspace

*ATTN

Attention

*F1-*F24

F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF7 key value (PF7)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP

5250 Help

*HLP3270

3270 Help Text (Display Active Keyboard Map)

*CLEAR

Clear Screen

*PRINT

Print Screen

*DSPATR

Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP   Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF8 key value (PF8)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
>   Print Screen

*DSPATR
>   Display Imbedded Attributes

*TEST Test Request

*DOWN
>   Roll Down

*UP   Roll Up

*NONE
>   No Assignment

*RESET
>   Error Reset

*SYSREQ
>   System Request

*BCKSPC
>   Record Backspace

*ATTN
>   Attention

*F1-*F24
>   F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
*   PF1 through PF12
*   PA1/PF1 through PF12
*   PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF9 key value (PF9)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
  5250 Help

*HLP3270
  3270 Help Text (Display Active Keyboard Map)

*CLEAR
  Clear Screen

*PRINT
  Print Screen

*DSPATR
  Display Imbedded Attributes

*TEST Test Request

*DOWN
  Roll Down

*UP Roll Up

*NONE
  No Assignment

*RESET
  Error Reset

*SYSREQ
  System Request

*BCKSPC
  Record Backspace

*ATTN
  Attention

*F1-*F24
  F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

Top

PF10 key value (PF10)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
  5250 Help

*HLP3270
  3270 Help Text (Display Active Keyboard Map)

*CLEAR
  Clear Screen

*PRINT
  Print Screen

*DSPATR
  Display Imbedded Attributes

*TEST Test Request

*DOWN
  Roll Down

*UP Roll Up

*NONE
  No Assignment

*RESET
  Error Reset

*SYSREQ
  System Request

*BCKSPC
  Record Backspace

*ATTN
  Attention

*F1-*F24
  F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF11 key value (PF11)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
5250 Help

*HLP3270
3270 Help Text (Display Active Keyboard Map)

*CLEAR
Clear Screen

*PRINT
Print Screen

*DSPATR
Display Imbedded Attributes

*TEST Test Request

*DOWN
Roll Down

*UP Roll Up

*NONE
No Assignment

*RESET
Error Reset

*SYSREQ
System Request

*BCKSPC
Record Backspace

*ATTN
Attention

*F1-*F24
F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF12 key value (PF12)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF13 key value (PF13)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
5250 Help

*HLP3270
3270 Help Text (Display Active Keyboard Map)

*CLEAR
Clear Screen

*PRINT
Print Screen

*DSPATR
Display Imbedded Attributes

*TEST Test Request

*DOWN
Roll Down

*UP Roll Up

*NONE
No Assignment

*RESET
Error Reset

*SYSREQ
System Request

*BCKSPC
Record Backspace

*ATTN
>   Attention

*F1-*F24
>   F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

*   PF1 through PF12
*   PA1/PF1 through PF12
*   PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF14 key value (PF14)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
>   5250 Help

*HLP3270
>   3270 Help Text (Display Active Keyboard Map)

*CLEAR
>   Clear Screen

*PRINT
>   Print Screen

*DSPATR
>   Display Imbedded Attributes

*TEST Test Request

*DOWN
>   Roll Down

*UP Roll Up

*NONE
>   No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF15 key value (PF15)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP   Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

Top

PF16 key value (PF16)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
>   Print Screen

*DSPATR
>   Display Imbedded Attributes

*TEST Test Request

*DOWN
>   Roll Down

*UP   Roll Up

*NONE
>   No Assignment

*RESET
>   Error Reset

*SYSREQ
>   System Request

*BCKSPC
>   Record Backspace

*ATTN
>   Attention

*F1-*F24
>   F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF17 key value (PF17)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

Top

PF18 key value (PF18)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF19 key value (PF19)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
5250 Help

*HLP3270
3270 Help Text (Display Active Keyboard Map)

*CLEAR
Clear Screen

*PRINT
Print Screen

*DSPATR
Display Imbedded Attributes

*TEST Test Request

*DOWN
Roll Down

*UP Roll Up

*NONE
No Assignment

*RESET
Error Reset

*SYSREQ
System Request

*BCKSPC
Record Backspace

*ATTN
Attention

*F1-*F24
F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF20 key value (PF20)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
5250 Help

*HLP3270
3270 Help Text (Display Active Keyboard Map)

*CLEAR
Clear Screen

*PRINT
Print Screen

*DSPATR
Display Imbedded Attributes

*TEST Test Request

*DOWN
Roll Down

*UP Roll Up

*NONE
No Assignment

*RESET
Error Reset

*SYSREQ
System Request

*BCKSPC
Record Backspace

*ATTN
Attention

*F1-*F24
F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF21 key value (PF21)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
   5250 Help

*HLP3270
   3270 Help Text (Display Active Keyboard Map)

*CLEAR
   Clear Screen

*PRINT
   Print Screen

*DSPATR
   Display Imbedded Attributes

*TEST Test Request

*DOWN
   Roll Down

*UP Roll Up

*NONE
   No Assignment

*RESET
   Error Reset

*SYSREQ
   System Request

*BCKSPC
   Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

Top

PF22 key value (PF22)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
    Error Reset

*SYSREQ
    System Request

*BCKSPC
    Record Backspace

*ATTN
    Attention

*F1-*F24
    F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PF23 key value (PF23)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
    5250 Help

*HLP3270
    3270 Help Text (Display Active Keyboard Map)

*CLEAR
    Clear Screen

*PRINT
    Print Screen

*DSPATR
    Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP   Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

* PF1 through PF12
* PA1/PF1 through PF12
* PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

Top

PF24 key value (PF24)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT

      Print Screen

*DSPATR

      Display Imbedded Attributes

*TEST Test Request

*DOWN

      Roll Down

*UP   Roll Up

*NONE

      No Assignment

*RESET

      Error Reset

*SYSREQ

      System Request

*BCKSPC

      Record Backspace

*ATTN

      Attention

*F1-*F24

      F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF1 key value (PA1PF1)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF2 key value (PA1PF2)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF3 key value (PA1PF3)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
5250 Help

*HLP3270
3270 Help Text (Display Active Keyboard Map)

*CLEAR
Clear Screen

*PRINT
Print Screen

*DSPATR
Display Imbedded Attributes

*TEST Test Request

*DOWN
Roll Down

*UP Roll Up

*NONE
No Assignment

*RESET
Error Reset

*SYSREQ
System Request

*BCKSPC
Record Backspace

*ATTN
Attention

*F1-*F24
F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF4 key value (PA1PF4)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF5 key value (PA1PF5)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
>   Attention

*F1-*F24
>   F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

*   PF1 through PF12
*   PA1/PF1 through PF12
*   PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF6 key value (PA1PF6)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
>   5250 Help

*HLP3270
>   3270 Help Text (Display Active Keyboard Map)

*CLEAR
>   Clear Screen

*PRINT
>   Print Screen

*DSPATR
>   Display Imbedded Attributes

*TEST Test Request

*DOWN
>   Roll Down

*UP Roll Up

*NONE
>   No Assignment

*RESET
    Error Reset

*SYSREQ
    System Request

*BCKSPC
    Record Backspace

*ATTN
    Attention

*F1-*F24
    F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF7 key value (PA1PF7)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
    5250 Help

*HLP3270
    3270 Help Text (Display Active Keyboard Map)

*CLEAR
    Clear Screen

*PRINT
    Print Screen

*DSPATR
    Display Imbedded Attributes

*TEST Test Request

*DOWN
    Roll Down

*UP   Roll Up

*NONE
    No Assignment

*RESET
    Error Reset

*SYSREQ
    System Request

*BCKSPC
    Record Backspace

*ATTN
    Attention

*F1-*F24
    F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF8 key value (PA1PF8)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
    5250 Help

*HLP3270
    3270 Help Text (Display Active Keyboard Map)

*CLEAR
    Clear Screen

*PRINT
      Print Screen

*DSPATR
      Display Imbedded Attributes

*TEST Test Request

*DOWN
      Roll Down

*UP   Roll Up

*NONE
      No Assignment

*RESET
      Error Reset

*SYSREQ
      System Request

*BCKSPC
      Record Backspace

*ATTN
      Attention

*F1-*F24
      F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF9 key value (PA1PF9)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
>5250 Help

*HLP3270
>3270 Help Text (Display Active Keyboard Map)

*CLEAR
>Clear Screen

*PRINT
>Print Screen

*DSPATR
>Display Imbedded Attributes

*TEST Test Request

*DOWN
>Roll Down

*UP Roll Up

*NONE
>No Assignment

*RESET
>Error Reset

*SYSREQ
>System Request

*BCKSPC
>Record Backspace

*ATTN
>Attention

*F1-*F24
>F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF10 key value (PA1PF10)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
    5250 Help

*HLP3270
    3270 Help Text (Display Active Keyboard Map)

*CLEAR
    Clear Screen

*PRINT
    Print Screen

*DSPATR
    Display Imbedded Attributes

*TEST Test Request

*DOWN
    Roll Down

*UP   Roll Up

*NONE
    No Assignment

*RESET
    Error Reset

*SYSREQ
    System Request

*BCKSPC
    Record Backspace

*ATTN
    Attention

*F1-*F24
    F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF11 key value (PA1PF11)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
5250 Help

*HLP3270
3270 Help Text (Display Active Keyboard Map)

*CLEAR
Clear Screen

*PRINT
Print Screen

*DSPATR
Display Imbedded Attributes

*TEST Test Request

*DOWN
Roll Down

*UP Roll Up

*NONE
No Assignment

*RESET
Error Reset

*SYSREQ
System Request

*BCKSPC
Record Backspace

*ATTN
Attention

*F1-*F24
F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA1-PF12 key value (PA1PF12)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
 5250 Help

*HLP3270
 3270 Help Text (Display Active Keyboard Map)

*CLEAR
 Clear Screen

*PRINT
 Print Screen

*DSPATR
 Display Imbedded Attributes

*TEST Test Request

*DOWN
 Roll Down

*UP Roll Up

*NONE
 No Assignment

*RESET
 Error Reset

*SYSREQ
 System Request

*BCKSPC
 Record Backspace

*ATTN
 Attention

*F1-*F24
 F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA2-PF1 key value (PA2PF1)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
 5250 Help

*HLP3270
 3270 Help Text (Display Active Keyboard Map)

*CLEAR
 Clear Screen

*PRINT
 Print Screen

*DSPATR
 Display Imbedded Attributes

*TEST Test Request

*DOWN
 Roll Down

*UP Roll Up

*NONE
 No Assignment

*RESET
 Error Reset

*SYSREQ
 System Request

*BCKSPC
 Record Backspace

*ATTN
>   Attention

*F1-*F24
>   F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

*   PF1 through PF12
*   PA1/PF1 through PF12
*   PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA2-PF2 key value (PA2PF2)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
>   5250 Help

*HLP3270
>   3270 Help Text (Display Active Keyboard Map)

*CLEAR
>   Clear Screen

*PRINT
>   Print Screen

*DSPATR
>   Display Imbedded Attributes

*TEST   Test Request

*DOWN
>   Roll Down

*UP   Roll Up

*NONE
>   No Assignment

*RESET
>   Error Reset

*SYSREQ
>   System Request

*BCKSPC
>   Record Backspace

*ATTN
>   Attention

*F1-*F24
>   F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA2-PF3 key value (PA2PF3)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
>   5250 Help

*HLP3270
>   3270 Help Text (Display Active Keyboard Map)

*CLEAR
>   Clear Screen

*PRINT
>   Print Screen

*DSPATR
>   Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

Top

PA2-PF4 key value (PA2PF4)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
>   Print Screen

*DSPATR
>   Display Imbedded Attributes

*TEST Test Request

*DOWN
>   Roll Down

*UP   Roll Up

*NONE
>   No Assignment

*RESET
>   Error Reset

*SYSREQ
>   System Request

*BCKSPC
>   Record Backspace

*ATTN
>   Attention

*F1-*F24
>   F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
*   PF1 through PF12
*   PA1/PF1 through PF12
*   PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA2-PF5 key value (PA2PF5)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
>   5250 Help

*HLP3270
>   3270 Help Text (Display Active Keyboard Map)

*CLEAR
>   Clear Screen

*PRINT
>   Print Screen

*DSPATR
>   Display Imbedded Attributes

*TEST Test Request

*DOWN
>   Roll Down

*UP Roll Up

*NONE
>   No Assignment

*RESET
>   Error Reset

*SYSREQ
>   System Request

*BCKSPC
>   Record Backspace

*ATTN
>   Attention

*F1-*F24
>   F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
* PF1 through PF12
* PA1/PF1 through PF12
* PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

Top

PA2-PF6 key value (PA2PF6)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
>   5250 Help

*HLP3270
>   3270 Help Text (Display Active Keyboard Map)

*CLEAR
>   Clear Screen

*PRINT
>   Print Screen

*DSPATR
>   Display Imbedded Attributes

*TEST   Test Request

*DOWN
>   Roll Down

*UP   Roll Up

*NONE
>   No Assignment

*RESET
>   Error Reset

*SYSREQ
>   System Request

*BCKSPC
>   Record Backspace

*ATTN
>   Attention

*F1-*F24
>   F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA2-PF7 key value (PA2PF7)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
5250 Help

*HLP3270
3270 Help Text (Display Active Keyboard Map)

*CLEAR
Clear Screen

*PRINT
Print Screen

*DSPATR
Display Imbedded Attributes

*TEST Test Request

*DOWN
Roll Down

*UP Roll Up

*NONE
No Assignment

*RESET
Error Reset

*SYSREQ
System Request

*BCKSPC
Record Backspace

*ATTN
Attention

*F1-*F24
F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

<div align="right">Top</div>

PA2-PF8 key value (PA2PF8)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
5250 Help

*HLP3270
3270 Help Text (Display Active Keyboard Map)

*CLEAR
Clear Screen

*PRINT
Print Screen

*DSPATR
Display Imbedded Attributes

*TEST Test Request

*DOWN
Roll Down

*UP Roll Up

*NONE
No Assignment

*RESET
Error Reset

*SYSREQ
System Request

*BCKSPC
Record Backspace

*ATTN
Attention

*F1-*F24
F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA2-PF9 key value (PA2PF9)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
     5250 Help

*HLP3270
     3270 Help Text (Display Active Keyboard Map)

*CLEAR
     Clear Screen

*PRINT
     Print Screen

*DSPATR
     Display Imbedded Attributes

*TEST Test Request

*DOWN
     Roll Down

*UP   Roll Up

*NONE
     No Assignment

*RESET
     Error Reset

*SYSREQ
     System Request

*BCKSPC
     Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

<div align="right">Top</div>

PA2-PF10 key value (PA2PF10)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:
- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA2-PF11 key value (PA2PF11)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP
> 5250 Help

*HLP3270
> 3270 Help Text (Display Active Keyboard Map)

*CLEAR
> Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN

Roll Down

*UP   Roll Up

*NONE

No Assignment

*RESET

Error Reset

*SYSREQ

System Request

*BCKSPC

Record Backspace

*ATTN

Attention

*F1-*F24

F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

PA2-PF12 key value (PA2PF12)

Each valid key or key sequence is a separate parameter which, except for restrictions as noted, can be assigned any function. If you do not specify a function for a particular key or key sequence, the function currently assigned to that key or key sequence remains the same.

The possible functions that can be assigned are:

*HELP

5250 Help

*HLP3270

3270 Help Text (Display Active Keyboard Map)

*CLEAR

Clear Screen

*PRINT
> Print Screen

*DSPATR
> Display Imbedded Attributes

*TEST Test Request

*DOWN
> Roll Down

*UP Roll Up

*NONE
> No Assignment

*RESET
> Error Reset

*SYSREQ
> System Request

*BCKSPC
> Record Backspace

*ATTN
> Attention

*F1-*F24
> F1 through F24 Function Keys

Restrictions:

*HELP, *HLP3270, and *RESET must be assigned to one of the following 12-key sets of function keys:

- PF1 through PF12
- PA1/PF1 through PF12
- PA2/PF1 through PF12

The *HELP, *HLP3270, and *RESET functions are required for 3270 display station device support to function properly. Since not all 3270 keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless these functions are **also** assigned to one of the three sets listed above.

It is recommended that *F1 and *SYSREQ both also be assigned to one of the three sets.

The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

Top

Examples

```
SETKBDMAP   PF1(*F1)  PF2(*F2)  PF3(*F3)
            PF4(*F4)  PF5(*HLP3270)  PF9(*HELP)
```

This command reassigns the keyboard primarily for an application that makes frequent use of the 5250 CF keys F1, F2, F3, F4. All other PF key sequences are set to the default shown on the command prompt. The above command is started in the program that started the application (thus tailoring the display station to whatever application is run).

Error messages

*ESCAPE Messages

CPF8701

Specified device &1 not a 3270 device type.

CPF8702

&1 function key not correctly defined.

CPF8703

Device &1 not ready.

Set Object Access (SETOBJACC)

The Set Object Access (SETOBJACC) command temporarily changes the speed of access to an object by bringing the object into a main storage pool or purging it from all main storage pools. An object can be kept main storage resident by selecting a pool for the object that has available space and does not have jobs associated with it. Repeated use of the command can cause a set of objects to be resident in a main storage pool.

Restrictions:

- You must have object operational (*OBJOPR) authority to all objects that are brought into a main storage pool or purged from all main storage pools.

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	*Qualified object name*	Required, Positional 1
	Qualifier 1: Object	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *USRLIBL, *CURLIB, *ALL, *ALLUSR	
OBJTYPE	Object type	*FILE, *PGM	Optional, Positional 2
POOL	Storage pool	*Element list*	Optional, Positional 3
	Element 1: Shared pool or subsystem name	*Name*, *JOB, *BASE, *SHRPOOL1, *SHRPOOL2, *SHRPOOL3, *SHRPOOL4, *SHRPOOL5, *SHRPOOL6, *SHRPOOL7, *SHRPOOL8, *SHRPOOL9, *SHRPOOL10, *SHRPOOL11, *SHRPOOL12, *SHRPOOL13, *SHRPOOL14, *SHRPOOL15, *SHRPOOL16, *SHRPOOL17, *SHRPOOL18, *SHRPOOL19, *SHRPOOL20, *SHRPOOL21, *SHRPOOL22, *SHRPOOL23, *SHRPOOL24, *SHRPOOL25, *SHRPOOL26, *SHRPOOL27, *SHRPOOL28, *SHRPOOL29, *SHRPOOL30, *SHRPOOL31, *SHRPOOL32, *SHRPOOL33, *SHRPOOL34, *SHRPOOL35, *SHRPOOL36, *SHRPOOL37, *SHRPOOL38, *SHRPOOL39, *SHRPOOL40, *SHRPOOL41, *SHRPOOL42, *SHRPOOL43, *SHRPOOL44, *SHRPOOL45, *SHRPOOL46, *SHRPOOL47, *SHRPOOL48, *SHRPOOL49, *SHRPOOL50, *SHRPOOL51, *SHRPOOL52, *SHRPOOL53, *SHRPOOL54, *SHRPOOL55, *SHRPOOL56, *SHRPOOL57, *SHRPOOL58, *SHRPOOL59, *SHRPOOL60, *PURGE	
	Element 2: Pool identifier	1-10	
MBR	Member	*Name*, ***FIRST**	Optional
MBRDATA	Member data	***BOTH**, *ACCPTH, *DATA	Optional

Object (OBJ)

Specifies the qualified name of the object to be brought into or deleted from main storage.

The name of the specified object can be qualified by one of the following library values:

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
>   The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*USRLIBL
>   Only the libraries in the user portion of the job's library list are searched.

*ALL   All libraries on the system are searched.

*ALLUSR
>   All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

```
#CGULIB      #DSULIB      #SEULIB
#COBLIB      #RPGLIB
#DFULIB      #SDALIB
```

>   Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered user libraries and are also searched:

```
QDSNX        QRCLxxxxx    QUSRIJS      QUSRVxRxMx
QGPL         QSRVAGT      QUSRINFSKR
QGPL38       QSYS2        QUSRNOTES
QMGTC        QSYS2xxxxx   QUSROND
QMGTC2       QS36F        QUSRPOSGS
QMPGDATA     QUSER38      QUSRPOSSA
QMQMDATA     QUSRADSM     QUSRPYMSVR
QMQMPROC     QUSRBRM      QUSRRDARS
QPFRDATA     QUSRDIRCL    QUSRSYS
QRCL         QUSRDIRDB    QUSRVI
```

>   1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
>   2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

library-name
>   Specify the name of the library to be searched.

The possible values are:

object-name
>   Specify the name of the object.

Object type (OBJTYPE)

Specifies the type of object to be brought into or deleted from main storage.

The possible values are:

*FILE   The object is a file.

*PGM   The object is a program.

Storage pool (POOL)

Specifies whether the object is brought into or purged from main storage.

The possible values are:

*JOB   The object is brought into the pool associated with the job.

*BASE
>   The object is brought into the base pool.

*SHRPOOLn
>   The object is brought into a a general-purpose shared pool. Valid values range from 1 through 10.

*PURGE
>   The object is purged from all pools. **Element 1: Subsystem**

subsystem
>   Specify a subsystem name. **Element 2: Pool Identifier**

pool-identifier
>   Specify a subsystem pool identifier.

*PURGE
>   The object is purged from all pools.

Member (MBR)

Specifies the database file member to be brought into or purged from main storage.

The possible values are:

*FIRST
>   The first member is selected.

file-member-name
>   Specify the member name.

Member data (MBRDATA)

Specifies the member data to be brought into or purged from main storage.

The possible values are:

*BOTH
>   All parts of the object are selected.

*ACCPTH
>   The file member's access path is selected.

*DATA
>   The file member's data is selected.

Examples

```
SETOBJACC   OBJ(OBJA)   OBJTYPE(*PGM)   POOL(*JOB)
```

This command brings a program named OBJA to the pool associated with the job in which the command is run.

Error messages

*ESCAPE Messages

CPF1858
>   The specified pool does not exist.

CPF1859
>   Use of an access path was requested but none exists.

CPF9855
>   File &1 in library &3 contains no members.

Set Program Information (SETPGMINF)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Set Program Information (SETPGMINF) command is used with the extended program model (EPM) languages to associate all the program objects in an application. The SETPGMINF command defines the application environment, based on the information you supply on the ROOTPGM, SUBPGM, and LIBFILE parameters. The C/400*, FORTRAN/400*, and Pascal languages work within the extended program model.

EPM programs that refer to external symbols in other EPM programs must be specified on the SETPGMINF command. External symbols are calls to other compilation units or external storage. You do not need to use this command if your program consists of only one compilation unit, or if the program calls only non-EPM programs. If the programs you specify on the ROOTPGM and SUBPGM parameters have not been successfully compiled, SETPGMINF fails.

Error messages for SETPGMINF

*ESCAPE Messages

PSE4017

>   Errors occurred in SETPGMINF command.

Top

Parameters

Keyword	Description	Choices	Notes
ROOTPGM	Root program	*Qualified object name*	Required, Positional 1
	Qualifier 1: Root program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SUBPGM	Sub-programs	Values (up to 200 repetitions): *Element list*	Optional, Positional 2
	Element 1: Program	*Qualified object name*	
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LIBFILE	Library information files	Values (up to 16 repetitions): *Element list*	Optional, Positional 3
	Element 1: File	*Qualified object name*	
	Qualifier 1: File	*Name*, ***SAME**, *NONE, *PASLIB, *FTNLIB, *CLIB	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LISTDETAIL	Listing detail	***NONE**, *BASIC, *FULL	Optional
PRTFILE	Print file	*Qualified object name*	Optional
	Qualifier 1: Print file	*Name*, **QSYSPRT**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Keyword	Description	Choices	Notes
RUNATTR	Run attributes	*Element list*	Optional
	Element 1: Maximum non-fatal errors	0-100, **20**, *NOMAX	
	Element 2: Fatal error severity	0-40, **40**	
	Element 3: External Type Checking	***YES**, *NO	
	Element 4: Computational Attributes	***LANG**, *ALL, *NONE	
PFROPT	Performance options	*Element list*	Optional
	Element 1: Access group storage	***NONE**, *ALL	
HEAPSIZE	Initial size of heap spaces	*Element list*	Optional
	Element 1: Dynamic storage heap	1024-16777216, **16000**, *NONE	
	Element 2: Static storage heap	1024-16777216, **32000**	
STACKSIZE	Initial size of auto storage	*Element list*	Optional
	Element 1:	1024-16777216, **16000**	
SSNATTR	Session file attributes	*Element list*	Optional
	Element 1: File size	8192-16000000, **32000**	
	Element 2: Buffer size	80-1024, **160**	
DBGOPT	Debug options	*Element list*	Optional
	Element 1: Debug initialization	***ON**, *OFF	

Root program (ROOTPGM)

Specifies the name of the program which will contain the environment definition necessary to create the EPM run-time environment. This program, called a default entry point or root program, contains the references to external symbols. In C, the default entry point is usually the program that contains a main() function. Pascal defines the main begin block of the program unit as the default entry point. FORTRAN defines the main program (the program defined on the PROGRAM statement) as the default entry point. *Change authority is required.

ROOTPGM is a required parameter.

program-name
>   The name of the EPM program object that contains the default entry point.

Possible library values are:

*LIBL   The system searches the library list for the root program object.

*CURLIB
>   The current library is searched. If you have not specified the current library, QGPL is used.

library-name
>   Enter the name of the library where the root program object is located.

Sub-programs (SUBPGM)

Specifies a list of all the program objects that you want to include in the run-time environment for your application. These programs can be either EPM or non-EPM program objects.

Up to 200 program objects can be specified on the SUBPGM parameter. You can specify more than 200 related program objects for 1 application if you nest SETPGMINF commands.

program-name

> Enter the name of the program object. You can specify up to 200 program objects on one SETPGMINF command.

Possible library values are:

*LIBL The system searches the library list for the library containing the program objects.

*CURLIB

> The name of the current library is used. If you have not specified the current library, QGPL is used.

library-name

> Enter the name of the library where the program object is located.

These SUBPGMs are sought at run-time, according to the library specifications used when you specify the SETPGMINF command. If you specify a specific library, or use *CURLIB, only that specific library is searched for the program object. If the library designated as the current library changes between the time you issue the SETPGMINF command and the time you run your program, not all of the program objects will be found and you will receive an error message.

Library information files (LIBFILE)

Specifies the library information file that contains information used to resolve any outstanding external references after all of the program objects identified on the ROOTPGM and SUBPGM parameters have been processed. Library information files are searched in the order that they are specified.

Several library information files are IBM-supplied. These files contain the EPM language library functions.

You can create and update your own library information file with the Extract Program Information (EXTPGMINF) command.

*SAME

> Use the same library information file as was used in the last environment definition. If this is the first time an environment is defined, the run-time library file for the language object specified on the ROOTPGM parameter is used. For example, if your ROOTPGM was compiled using the C/400 compiler, the default library information file is *CLIB.

*NONE

> No library information file is used.

*PASLIB

> The Pascal run-time library information file is used.

*CLIB The C/400 run-time library information file is used.

*FTNLIB

> The FORTRAN/400 run-time library information file is used.

file-name
> Enter the name of the library information file. This library information file must exist. To create a library information file, use the EXTPGMINF command.

The possible library values are:

*LIBL   The system searches the library list for the specified library information file.

*CURLIB
> The current library is used to locate the library information file. If you have not specified the current library, QGPL is used.

library-name
> Enter the name of the library that contains the library information file.

Listing detail (LISTDETAIL)

Specifies whether a SETPGMINF listing is created. The listing will have the same name as the program specified on the ROOTPGM parameter and is directed to the library and file specified on the PRTFILE parameter.

The PRTFILE parameter is not displayed unless you request a listing (*BASIC or *FULL) on the LISTDETAIL parameter.

*NONE
> A listing is not created.

*BASIC
> A listing is created that includes all of the symbol references and definitions that result from the programs specified on the ROOTPGM and SUBPGM parameters.

*FULL   A listing is created that includes all of the symbol references and definitions for the entire application. This includes the symbol references and definitions for all the programs specified on the LIBFILE parameter that are not explicitly referenced in your application.

Print file (PRTFILE)

Specifies the name and library of the printer file where the SETPGMINF listing is directed. The file should have a minimum length of 132. If you specify a file with a record length of less than 132, information may be lost.

This parameter does not appear on the prompting display unless you change the default value on the LISTDETAIL parameter to *BASIC or *FULL.

QSYSPRT
> The SETPGMINF listing is placed in the file QSYSPRT.

file-name
> Enter the name of the file where the SETPGMINF listing is to be placed.

The possible library values are:

*LIBL   The system searches the library list.

*CURLIB
> The name of the current library is used. If you have not specified the current library, QGPL is used.

library-name
> Enter the name of the library where the file is stored.

Run attributes (RUNATTR)

Specifies the number of times the non-fatal error counter is incremented before processing ends, and the message severity-level that is interpreted as a fatal error at run-time. A fatal error is an error that stops your application from running. You can also specify whether you want external type checking to take place at program run-time. The computational attributes field allows you to set the attributes for controlling floating point operations in the run-time environment.

20
> The non-fatal error counter is incremented up to 20 times before the processing of your application ends.

*NOMAX
> The non-fatal error counter has no limit and will not stop the processing of your application.

counter-number
> The number of non-fatal errors that are allowed before processing ends.

40   A message with a severity-level of 40 or higher is interpreted as a fatal error.

severity-level
> The message severity-level that is interpreted as a fatal error. If a severity-level of 0 is specified, your application stops running if any errors occur.

*YES   External type checking is performed at program run-time.

*NO   External type checking is not performed at program run-time. Any warning messages that occur as a result of checking errors are not issued when you run your program.

*LANG
> Computational attributes are set according to the semantics of the EPM language.

*ALL   Floating point exceptions are disabled or masked.

*NONE
> Floating point exceptions are enabled or unmasked.

Performance options (PFROPT)

Specifies whether the space allocated for static, automatic, and dynamic program variables is part of the process access group.

*NONE
> The storage spaces created are not part of the process access group (PAG).

*ALL   The storage spaces created are part of the process access group (PAG). This option can improve the performance of programs that use limited program variable storage, and are run on a non-dedicated system. Information on performance tools and CL commands you can use to display and analyze the PAG can be found in the Performance Tools Guide.

Initial size of heap spaces (HEAPSIZE)

Specifies the <u>initial</u> size of the heap spaces for the dynamic and static storage heap. The heap size specified will expand as your application progresses.

The dynamic storage heap consists of the storage that you have allocated in your application with the C/400 memory routines (malloc, calloc, realloc) or the NEW procedure in Pascal.

The static storage heap consists of the storage that you have allocated in your application with static and external static variables.

16000 The initial size of the dynamic storage heap is 16000 bytes.

*NONE
    No initial dynamic storage is allocated. Use *NONE only if you do not use any EPM memory management functions in your application. If you specify *NONE and use any of the memory functions, you will get pointer exceptions and your application will stop processing.

dynamic-storage-size
    Enter the initial size of the dynamic storage heap. The initial size can be between 1024 and 16777216 bytes.

32000 The initial size of the static storage heap is 32000 bytes.

static-storage-size
    Enter the initial size of the static storage heap. The initial size can be between 1024 and 16777216 bytes. If you use static variables extensively in your application your performance can improve if you specify a larger storage heap than the default.

Initial size of auto storage (STACKSIZE)

Specifies the <u>initial</u> size of the automatic storage stack. The stack size specified will expand as your application progresses.

The automatic storage stack consists of the storage that you have allocated in your application with automatic variables.

16000 The initial size of the automatic storage stack is 16000 bytes.

automatic-storage-size
    Enter the initial size of the automatic storage stack. The initial size can be between 1024 and 16777216 bytes. If your application contains a large number of recursive calls, you should increase the initial size of the stack.

Session file attributes (SSNATTR)

Specifies the file session attributes for the file and buffer size.

32000 The default size for the file is 32000 bytes.

file-size
    Specify a value for the file size between 8192 and 16000000 bytes. If your file exceeds the size you specify, the information at the top rolls off the display. This information is lost.

160 The default size for the buffer is 160 bytes.

buffer-size
>   Specify a value for the buffer size between 80 and 1024 bytes.

Debug options (DBGOPT)

Specifies the use of the extended program model (EPM) debug tool at run-time, if server debug mode is active. To activate server debug mode, enter the STRDBG command.

*ON EPM debug starts at run-time if server debug mode is active.

*OFF EPM debug will not start at run-time. You can use server debug to debug your programs, but EPM language variable names, statement numbers, and debugging commands are not available.

Examples

None

Error messages

*ESCAPE Messages

PSE4017
>   Errors occurred in SETPGMINF command.

Set Tape Category (SETTAPCGY)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Set Tape Category (SETTAPCGY) command sets the category for a tape device in a specified media library device. The system automatically loads cartridges from the specified category in the order specified on the cartridge order (CTGORDER) parameter.

Parameters

Keyword	Description	Choices	Notes
DEV	Library device	*Name*	Required, Positional 1
OPTION	Option	*MOUNTED, *DEMOUNTED, *ASSIGN, *RELEASE	Required, Positional 2
CGY	Category	Single values: *SHARE400 Other values: *Element list*	Optional
	Element 1: Category name	*Character value*, ***NOSHARE**, *IPL, *NL, *CNV	
	Element 2: Category system	*Character value*, ***CURRENT**	
CTGORDER	Cartridge order	***SEQ**, *NEXTAVAIL	Optional
TGTCGY	Target category	Single values: *SHARE400 Other values: *Element list*	Optional
	Element 1: Category name	*Character value*, ***CGY**, *NOSHARE, *IPL, *NL	
	Element 2: Category system	*Character value*, ***CURRENT**	
MNTID	Mount identifier	*Name*, ***NONE**	Optional

Library device (DEV)

Specifies the media library device for which the category is set.

This is a required parameter.

name   Specify the name of the media library device.

Option (OPTION)

Specifies which option is set for the specified media library device.

This is a required parameter.

*MOUNTED

The category specified on the CGY parameter is considered mounted and any tape operation with VOL(*MOUNTED) specified uses the tape cartridges from the category that is mounted.

*DEMOUNTED

The category specified on the CGY parameter is no longer mounted. The use of VOL(*MOUNTED) is not valid for the media library device.

*ASSIGN

The mounted category session specified on the MNTID parameter is assigned to the job issuing the SETTAPCGY command. The mounted category session being assigned must have been previously mounted and released.

*RELEASE

The mounted category session assigned to the job issuing the SETTAPCGY command is released and is available for another job to assign.

Category (CGY)

Specifies the category to be mounted.

Single values

*SHARE400

The cartridge identifier can be shared by all iSeries systems that are attached to the library device. The cartridge identifiers are mounted in the order specified in the CTGORDER parameter.

Element 1: Category name

*NOSHARE

The cartridge identifiers cannot be shared with other systems that are attached to the same device. The cartridge identifiers are mounted in the order specified in the CTGORDER parameter.

*IPL The cartridge identifiers can be used for an alternate initial program load (IPL) of the system. The cartridge identifiers are mounted in the order specified in the CTGORDER parameter.

*NL The cartridge is used as a non-labeled tape. The cartridge identifiers are mounted in the order specified in the CTGORDER parameter.

*CNV The cartridge identifier is used from the special convenience category. The cartridge identifiers are mounted in the order specified in the CTGORDER parameter.

character-value

Specify the name of a user-defined category. The cartridge identifiers in the category specified are mounted in the order specified in the CTGORDER parameter.

Element 2: Category system

The second part of the parameter specifies the name of the system to which the category belongs. The system name is obtained from the pending system name field of a Display Network Attributes (DSPNETA) command. If there is no pending system name, the current system name attribute is used.

```
* * * Attention * * *
If a system name is changed, the tape cartridges in
library devices that have the attribute of the system
name before it was changed are no longer valid.
```

*CURRENT

The category belongs to the system currently running the command.

character-value
> Specify the name of the system to which the category belongs.

Cartridge order (CTGORDER)

Specifies the order in which the cartridges are mounted.

*SEQ The cartridges are mounted in the order they were added or changed to the category specified. The operation ends if the next cartridge in the sequential order is not available.

*NEXTAVAIL
> The cartridges are mounted in a sequential order, but if a cartridge in the order is not available, the next available cartridge is used.

Target category (TGTCGY)

Specifies the category to which a tape cartridge is changed after it is used. This parameter can be useful when a scratch category is set for use during a save operation. Each cartridge is automatically changed to the specified target category after it is used.

Single values

*SHARE400
> The cartridge identifiers are changed to the *SHARE400 category.

Element 1: Category name

*CGY The cartridges remain in the category specified on the CGY parameter.

*NOSHARE
> The cartridge identifiers are changed to the *NOSHARE category.

*IPL The cartridge identifiers are changed to the *IPL category.

*NL The cartridge identifiers are changed to the *NL category.

character-value
> Specify the name of a user-defined category. The cartridge identifiers are changed to the specified user-defined category.

Element 2: Category system

The second part of this parameter specifies the name of the system to which the target category belongs. The system name is obtained from the pending system name field of the Display Network Attributes (DSPNETA) command. If there is no pending system name, the current system name attribute is used.

* * * Attention * * *
```
If a system name is changed, the tape cartridges in
library devices that have the attribute of the system
name before it was changed are no longer valid.
```

*CURRENT
> The system currently running the command is used.

character-value
> Specify the name of the system that the category belongs to.

Mount identifier (MNTID)

Specifies the identifier by which the mounted category session will be known. This parameter is valid only when OPTION(*MOUNTED) or OPTION(*ASSIGN) is specified.

*NONE
> This mounted category session is not assigned to any job and will be used by the first job that issues a command to the media library with a volume identifier of *MOUNTED. The mount identifier *NONE cannot be specified with OPTION(*ASSIGN).

name Specify a unique name to identify the mounted category session. A mounted category session known by this mount identifier is created and assigned when the category is mounted. This identifier is then used when assigning the session to another job. The session known by this identifier is deleted when the category is demounted. The mount identifier can then be reused when mounting another category to the media library.

Examples

Example 1: Using Mounted Category Without a Mount Identifier

```
SETTAPCGY  MLB(LIB01)  OPTION(*MOUNTED)  CGY(*NOSHARE *CURRENT)
           CTGORDER(*SEQ)  TGTCGY(*IPL)
```

This command sets the tape category to *NOSHARE for a resource in media library device LIB01 on the system currently running this command. Each cartridge that is used is changed to the *IPL category. The order in which the cartridges are used is the exact order in which they were added to or changed in the category. As each cartridge is used and unloaded from the resource, the system automatically chooses and loads the next sequential cartridge from the *NOSHARE category.

Example 2: Using Mounted Category with a Mount Identifer

This command sets the tape category to *NOSHARE for a resource in media library device LIB01 on the system currently running this command. The mounted category session is identified by the mount identifier DAILY. Each cartridge that is used is changed to the DAILY1 category. The order in which the cartridges are used is the exact order in which they were added to or changed in the category. As each cartridge is used and unloaded from the resource, the system automatically chooses and loads the next sequential cartridge from the *NOSHARE category.

Example 3: Releasing a Category Session

```
SETTAPCGY  MLB(LIB01)  OPTION(*RELEASE)
```

This command releases the category session assigned to the job issuing the command. The category is still set to a resource in media library device LIB01 and is available for another job to assign.

Example 4: Assigning a Category Session

```
SETTAPCGY  MLB(LIB01)  OPTION(*ASSIGN)  MNTID(DAILY)
```

This command assigns the mounted category session identified by the mount identifier DAILY to the job issuing the command.

Example 5: Demounting a Mounted Category

```
SETTAPCGY  MLB(LIB01)  OPTION(*DEMOUNTED)
```

This command demounts the mounted category from a resource in media library device LIB01. The mount identifier DAILY is now available to use to name another mounted category session.

Error messages

ESCAPE Messages

CPF6711
> Command not allowed

CPF6712
> Category &4 not mounted.

CPF6713
> Category not demounted.

CPF6745
> Device &1 not a media library device.

CPF67A6
> Category does not exist

CPF67AD
> Category not assigned.

CPF67AE
> Category not released.

CPF67E4
> Library device function not successful

CPF9814
> Device &1 not found.

CPF9825
> Not authorized to device &1.

Set Upgrade Environment (SETUPGENV)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Set Upgrade Environment (SETUPGENV) command prompts the user for information that is required to help the user plan for and perform the upgrade. The information that is gathered is stored in library QUPGRADE.

Parameters

Keyword	Description	Choices	Notes
UPGENV	Upgrade environment	*Character value*, *NEW, *UPGENVID	Required, Positional 1
TGTRLS	Target release	*Character value*	Optional, Positional 2
TGTPRC	Target processor	*Character value*, ***DFT**	Optional, Positional 3
MERGE	Merge with Advanced/36	***NO**, *YES	Optional, Positional 4
UPGENVID	Upgrade environment identifier	*Name*, *SRLNBR	Optional, Positional 5

Upgrade environment (UPGENV)

Specifies whether a new upgrade environment is being created or if you want to work with an existing environment.

*NEW Specifies a new upgrade environment is being created.

*UPGENVID
   Specifies that you want to work with an existing upgrade environment. The identifier of the existing upgrade environment is specified in the upgrade environment identifier field.

Target release (TGTRLS)

Specifies the target release to which you intend to upgrade. This field is only valid when the Upgrade Environment (UPGENV) value is *NEW.

target-release
   Specify the release level in the format VxRxMx. Valid values depend on the current version, release, and modification level, and they change with each new release.

© Copyright IBM Corp. 1998, 2006

743

Target processor (TGTPRC)

Specify the feature code of the iSeries processor you expect for your target system. This field is only valid when the Upgrade Environment (UPGENV) value is *NEW.

The processors listed are the valid target processors for upgrading your source iSeries system. Other valid processors may be entered but the upgrade would represent an unsupported upgrade.

*DFT   Specifies the smallest processor necessary to provide similar performance to that of the source system.

target-processor
> Specifies the four digit feature code of the target processor. Valid values depend on the source and target releases.

<div align="right">Top</div>

Merge with Advanced/36 (MERGE)

Indicates whether System/36 operating system (SSP) will be running as guest on the target system. This field is only valid when the Upgrade Environment (UPGENV) value is *NEW

*NO   Indicates this system will not be merged with a System/36

*YES   Indicates this system will be merged with a System/36

<div align="right">Top</div>

Upgrade environment identifier (UPGENVID)

Specifies the identifier associated with an upgrade environment. You may develop several upgrade environments as you are planning the order for your upgrade. Each environment may use different values and upgrade methods to help you decide which upgrade method and approach is best for you and your business. Only one upgrade environment will be used for upgrade preparation and the upgrade.

*SRLNBR
> Specifies the default name for the upgrade environment. The upgrade environment identifier for this environment will be the serial number of this system appended to the letter 'Q', for example, Q0100A12.

upgrade-environment-identifier
> Specifies a character name as the upgrade environment identifier.

<div align="right">Top</div>

Examples

None

<div align="right">Top</div>

Error messages

Unknown

<div align="right">Top</div>

Sign Off (SIGNOFF)

Where allowed to run: Interactive environments (*INTERACT
*IPGM *IREXX *EXEC)
Threadsafe: No

The Sign Off (SIGNOFF) command ends an interactive job or causes all jobs in a group to end. You enter this command to sign off at a work station.

Restrictions:

1. This command is valid only in an interactive job.
2. If the SIGNOFF command is issued in a CL program, all subsequent commands in the CL program are bypassed.

Top

Parameters

Keyword	Description	Choices	Notes
LOG	Job log	***NOLIST**, *LIST	Optional, Positional 1
DROP	Drop line	***DEVD**, *YES, *NO	Optional, Positional 2
ENDCNN	End connection	***NO**, *YES	Optional, Positional 3

Top

Job log (LOG)

Specifies whether the job log for this interactive job is deleted or is included in the job's spooled output for printing. This entry takes precedence over the log value specified for the job itself.

*NOLIST
    The information in the job log, which has already been displayed throughout the job, is no longer needed and is being deleted.

*LIST The job log is spooled for printing, along with the job's remaining spooled output, if any.

Top

Drop line (DROP)

Specifies, for switched lines only, whether the switched line attached to the work station is disconnected (dropped) if no other work stations on the same line are signed on. This parameter is ignored if the work station is attached to a nonswitched line.

*DEVD
    The value specified for the **Drop line (DROP)** parameter of the work station's device description is assumed.

*YES The switched line is disconnected when the job is ended if no other work stations on the line are signed on.

*NO The switched line is not disconnected when the job is ended.

End connection (ENDCNN)

Specifies whether to end the connection to the current system. Ending the connection allows the user to bypass the sign-on display of the target system and return to the source system. For communication functions that do not support this option, this parameter is ignored.

*NO The connection does not end. The sign-on display of the target system is shown.

*YES The connection ends and the user is returned to the source system. No sign-on screen or error messages are shown from the target system.

Examples

Example 1: Signing Off and Ending an Interactive Job

```
SIGNOFF
```

This command signs off the user of the work station and ends the interactive job. The switched line is dropped only if specified in the device description of this work station and if no other work station on this line is active. An end-of-job message that gives the job start and stop times is written in the job's log.

Example 2: Printing the Job Log

```
SIGNOFF   LOG(*LIST)  DROP(*NO)
```

This command ends the interactive job, but the switched line is not released. The job log is printed with the job's spooled output.

Example 3: Signing Off and Ending the Connection

```
SIGNOFF   ENDCNN(*YES)
```

This command ends the connection and transfers the user back to the source system.

Error messages

None

Select Command (SLTCMD)

The Select Command (SLTCMD) command allows you to display a list of commands from one or more libraries. From the Select Commands display, you can select a command that you want to prompt.

Restrictions:

- Only the libraries to which you have use (*USE) authority will be searched.
- Only the commands to which you have some authority will be shown on the display.
- To prompt a listed command, you must have *USE authority to the command.

Top

Parameters

Keyword	Description	Choices	Notes
CMD	Command	*Qualified object name*	Required, Positional 1
	Qualifier 1: Command	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB, *USRLIBL, *ALLUSR, *ALL	

Top

Command (CMD)

Specifies the commands to be shown on the Select Command display.

This is a required parameter.

Qualifier 1: Message queue

*ALL   All commands are shown.

generic-name
>   Specify the generic name of the commands to be shown. A generic name is a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, all commands that have names with the same prefix as the generic name are shown.

name   Specify the name of the command to be shown.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched. All objects in these libraries with the specified object name are shown.

*CURLIB
>   The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched.

*USRLIBL
>   If a current library entry exists in the library list for the current thread, the current library and the

libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched.

*ALLUSR

All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

```
#CGULIB     #DSULIB     #SEULIB
#COBLIB     #RPGLIB
#DFULIB     #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered user libraries and are also searched:

```
QDSNX       QRCLxxxxx   QUSRIJS     QUSRVxRxMx
QGPL        QSRVAGT     QUSRINFSKR
QGPL38      QSYS2       QUSRNOTES
QMGTC       QSYS2xxxxx  QUSROND
QMGTC2      QS36F       QUSRPOSGS
QMPGDATA    QUSER38     QUSRPOSSA
QMQMDATA    QUSRADSM    QUSRPYMSVR
QMQMPROC    QUSRBRM     QUSRRDARS
QPFRDATA    QUSRDIRCL   QUSRSYS
QRCL        QUSRDIRDB   QUSRVI
```

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).

2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

*ALL   All libraries in the system, including QSYS, are searched.

name   Specify the name of the library to be searched.

<div align="right">Top</div>

Examples

```
SLTCMD   CMD(QSYS/*ALL)
```

This command shows a list of all commands in library QSYS. The option to prompt and run commands is available.

<div align="right">Top</div>

Error messages

None

<div align="right">Top</div>

Send Break Message (SNDBRKMSG)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Send Break Message (SNDBRKMSG) command is used to send an immediate message to one or more work station message queues. An immediate message is a message that is not predefined and is not stored in a message file. The command causes the message to be delivered always in break mode. The DSPMSG display is shown for the message when it is received, regardless of the setting of the message queue's delivery mode, severity, and break handling program. However, the message may not be displayed in some cases, depending on the BRKMSG job attribute. This command is primarily intended for the system operator's use.

Restrictions:

1. This command can be used to send break messages to work station message queues only.
2. This command cannot send inquiry messages (specified by MSGTYPE(INQ)) to multiple work stations.

Parameters

Keyword	Description	Choices	Notes
MSG	Message text	*Character value*	Required, Positional 1
TOMSGQ	To work station message queue	Single values: *ALLWS Other values (up to 50 repetitions): *Qualified object name*	Required, Positional 2
	Qualifier 1: To work station message queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**	
MSGTYPE	Message type	***INFO**, *INQ	Optional, Positional 3
RPYMSGQ	Message queue to get reply	*Qualified object name*	Optional
	Qualifier 1: Message queue to get reply	*Name*, **QSYSOPR**	
	Qualifier 2: Library	*Name*, ***LIBL**	
CCSID	Coded character set ID	1-65535, *HEX, ***JOB**	Optional

Message text (MSG)

Specifies the immediate message that is being sent. The text must be enclosed in apostrophes if it contains blanks or other special characters. A maximum of 512 characters can be specified.

Coded Character Set Identifier (CCSID) Considerations

The text supplied for the MSG parameter is assumed to be in the CCSID of the job running this command unless a coded character set identifier is supplied for the CCSID parameter. For more

information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

This is a required parameter.

To work station message queue (TOMSGQ)

Specifies one or more work station message queues to which the break message is sent. Only the names of work station message queues can be specified and only *LIBL or QSYS can be specified for the library value.

Single values

*ALLWS
> The break message is sent to all work station and Personal Computer message queues. *ALLWS cannot be specified if *INQ is specified for the **Message type (MSGTYPE)** parameter.

Qualifier 1: To work station message queue

message-queue-name
> Specify the name of the message queue to which the break message is to be sent.

Qualifier 2: Library

*LIBL All libraries in the library list for the current thread are searched until the first match is found.

library-name
> Specify the library where the message queue is located.

Message type (MSGTYPE)

Specifies the type of message that is sent in break mode. Only informational or inquiry message types can be specified. Inquiry messages may require a response.

*INFO
> An information only message is sent in break mode.

*INQ An inquiry message is sent in break mode; the work station receiving the message is expected to reply to it. An inquiry message cannot be sent to multiple queues with only one command.

Message queue to get reply (RPYMSGQ)

Specifies, only if an inquiry message is sent, the message queue that the work station user's reply is sent to.

Qualifier 1: Message queue to get reply

QSYSOPR
> The replies to the break message are sent to the system operator's message queue, QSYSOPR.

message-queue-name
>   Specify the name of the message queue to which a reply to the break message is sent. Only a user or work station message queue can be specified.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

library-name
>   Specify the library where the message queue is located.

Coded character set ID (CCSID)

Specifies the coded character set identifier (CCSID) that the specified message text is in. The text supplied by the MSG parameter is assumed to be in the CCSID supplied by this parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

*JOB   The message text is assumed to be in the CCSID of the job running this command.

*HEX   The message text is not converted. CCSID 65535 is used.

coded-character-set-identifier
>   Specify a valid CCSID in which you want your message text to be considered in. Valid values range from 1 through 65535. See the Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for a list of valid CCSID values. This command will validate the CCSID.

Examples

Example 1: Sending a Message

```
SNDBRKMSG   MSG('Inventory application shuts down at 4:00 PM.')
```

This command sends the message 'The inventory application shuts down at 4:00 pm today.' to all work station message queues. If the work station is signed on, the message will be delivered in break mode regardless of the delivery attribute setting of those message queues. The message is also added to the work station message queues of those work stations that are not signed on.

Example 2: Sending an Immediate Message

```
SNDBRKMSG   MSG('Your printed output is ready.')
            TOMSGQ(GEORGEMSGQ)
```

This example shows a typical use of the SNDBRKMSG command by the system operator to send an immediate message to a work station user.

Error messages

*ESCAPE Messages

CPF2428
>   Message queue parameter is not valid.

CPF2469

Error occurred when sending message&1.

CPF247E

CCSID &1 is not valid.

CPF9838

User profile storage limit exceeded.

Top

Send Distribution (SNDDST)

Where allowed to run: All environments (*ALL)
Threadsafe: No

The Send Distribution (SNDDST) command allows you to send a distribution to a user, to a list of users, or to a distribution list.

Restrictions:

- If you are working on behalf of another user, you must have been granted permission to work for that user through the Grant User Permission (GRTUSRPMN) command.
- You need to be enrolled in the system distribution directory.
- You cannot request personal distribution if you are working for another user.

Parameters

Keyword	Description	Choices	Notes
TYPE	Information to be sent	*MSG, *DOC, *FILE, *IDP, *DSTID, *LMSG	Required, Positional 1
TOUSRID	Recipient	Values (up to 300 repetitions): *Element list*	Optional, Positional 2
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	
	Element 3: Recipient type	***PRI**, *CC, *BCC	
TOINTNET	Internet Recipient	Values (up to 300 repetitions): *Element list*	Optional
	Element 1: Internet address	*Character value*, ***NONE**	
	Element 2: Recipient type	***PRI**, *CC, *BCC	
DSTD	Description	*Character value*	Optional, Positional 3
MSG	Message	*Character value*, ***NONE**, *DSTIDMSG	Optional
LONGMSG	Long Message	*Character value*, ***NONE**	Optional
CFMDEL	Confirmation of delivery	***NO**, *YES	Optional
SENSITIV	Sensitivity	***NONE**, *PRIVATE, *PERSONAL, *CONFIDENTIAL	Optional
PERSONAL	Personal	***NO**, *YES	Optional
IMPORTANCE	Content importance	***NORMAL**, *LOW, *HIGH	Optional
PTY	Priority	***NORMAL**, *HIGH, *LOW	Optional
USRID	User identifier	Single values: ***CURRENT** Other values: *Element list*	Optional
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	
DOCFILE	Document file	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Document file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DOCMBR	Document member	*Name*, ***FIRST**	Optional

Keyword	Description	Choices	Notes
DOCTYPE	Document type	2-65535, ***DFT**, *FFT, *RFT	Optional
SNDFMT	Send format	***NOCHG**, *NOTE, *FINALFORM	Optional
SYSCOD	System code	*Character value*, ***DFT**	Optional
DOCCHRID	Document character identifier	Single values: ***SYSVAL**, *DEVD Other values: *Element list*	Optional
	Element 1: Graphic character set	*Integer*	
	Element 2: Code page	*Integer*	
DSTID	Distribution identifier	*Character value*, ***NONE**	Optional
DSTIDEXN	Distribution ID extension	0-99, ***NONE**	Optional
DOC	Document	*Character value*, *DOCID	Optional
FLR	In folder	*Character value*, ***NONE**	Optional
DOCID	Document identifier	*Character value*, ***NONE**	Optional
IDPFILE	Profile file	Single values: ***NONE**, *DOCFILE, *DSTIDIDP Other values: *Qualified object name*	Optional
	Qualifier 1: Profile file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
IDPMBR	Profile member	*Name*, ***FIRST**	Optional
DOCD	Document description	*Character value*, ***DFT**, *DSTD	Optional
AUTHOR	Author	Single values: ***NONE**, *USRID Other values (up to 50 repetitions): *Character value*	Optional
DOCCLS	Document class	*Character value*, ***NONE**	Optional
KWD	Keyword	Single values: ***NONE** Other values (up to 50 repetitions): *Character value*	Optional
SUBJECT	Subject	Single values: ***NONE**, *DOCD Other values (up to 50 repetitions): *Character value*	Optional
DOCDATE	Document date	*Date*, ***NONE**, *CURRENT	Optional
FILCAB	File cabinet location	*Character value*, ***NONE**	Optional
CPYLST	Copy list	Single values: ***NONE** Other values (up to 50 repetitions): *Character value*	Optional
EXPDATE	Expiration date	*Date*, ***NONE**	Optional
CRTDATE	Creation date	*Date*, ***NONE**, *CURRENT	Optional
REFERENCE	Reference	*Character value*, ***NONE**	Optional
ACTDATE	Action due date	*Date*, ***NONE**, *CURRENT	Optional
RPYDATE	Reply requested	*Element list*	Optional
	Element 1: Date	*Date*, ***NONE**, *CURRENT, *ANY	
	Element 2: Time	*Time*, ***ENDOFDAY**	
STATUS	Document status	*Character value*, ***NONE**	Optional
CMPDATE	Completion date	*Date*, ***NONE**, *CURRENT	Optional
PROJECT	Project	*Character value*, ***NONE**	Optional
DOCLANGID	Language ID	*Character value*, ***JOB**	Optional
DOCCNTRYID	Country or region ID	*Character value*, ***JOB**	Optional
ALWALTRCP	Allow alternate recipient	***YES**, *NO	Optional
DISCLOSE	Disclose recipient	***YES**, *NO	Optional
ALWX400CNV	Allow X.400 conversion	***YES**, *NO	Optional

Keyword	Description	Choices	Notes
AUTUSR	Authorizing user	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	
DSTEXPDATE	Distribution expiry indicator	*Element list*	Optional
	Element 1: Date	*Date*, ***NONE**	
	Element 2: Time	*Time*, ***ENDOFDAY**	
CMDCHRID	Command character identifier	Single values: ***SYSVAL**, **DEVD* Other values: *Element list*	Optional
	Element 1: Graphic character set	*Integer*	
	Element 2: Code page	*Integer*	

Information to be sent (TYPE)

Specifies the type of information that is sent and the parameters that are valid on this command.

*MSG Only the message specified on the **Message** prompt (MSG parameter) is sent.

*DOC The document specified on the **Document** prompt (DOC parameter) or the **Document identifier** prompt (DOCID parameter) is sent. The user must have authority for the document before it can be sent.

*FILE The database file specified on the **Document file** prompt (DOCFILE parameter) and the **Document member** prompt (DOCMBR parameter) is sent. The database file is sent without any changes. The user must have authority for the database file before it can be sent.

*IDP The interchange document profile (IDP) that is sent is specified on the **Profile file** prompt (IDPFILE parameter) and the **Profile member** prompt (IDPMBR parameter), or is specified in the document profile built by this command.

*DSTID
The mail entry that is identified by the distribution ID is distributed. The distribution ID is called the distribution document name.

*LMSG
The text specified on the Long Message prompt (LONGMSG parameter) is sent as an final-form text document (FFTDCA) note. LONGMSG allows up to 5000 characters, which is about one page of text. The "details" parameters (Subject, Reference, Author, Due Date, etc.) are supported with TYPE(*LMSG). The TOINTNET parameter is also supported with TYPE(*LMSG).

Recipient (TOUSRID)

Specifies one of the following:

- The user ID and address of zero or more users to whom the distribution is being sent
- The distribution list name of zero or more distribution lists containing the user ID and address of one or more users to whom the distribution is being sent.

A combination of user IDs and distribution lists can be used on the same command. Up to 300 user IDs and addresses can be specified.

If no user ID and address are specified for the TOUSRID parameter, an internet address must be specified on the TOINTNET parameter.

The possible User ID or List ID values are:

user-ID
> Specify the user ID of the user to whom the distribution is sent.

list-ID
> Specify the distribution list of users to whom the distribution is sent.

The possible User Address or List Qualifier values are:

user-address
> Specify the user address of the user (specified in Element 1) to whom the distribution is sent.

list-qualifier
> Specify the distribution list qualifier of users (specified in Element 1) to whom the distribution is sent.

The possible Recipient Type values are:

*PRI   The user or distribution list is the primary recipient of the distribution.

*CC   The user or distribution list is receiving a copy of the distribution sent to the primary recipient. However, this copy recipient is not identified on the distribution as a receiver on the distribution.

*BCC   The user or distribution list is receiving a copy of the distribution. However, this copy recipient is not identified on the distribution as a receiver on the distribution.

Internet Recipient (TOINTNET)

Specifies the internet address of one or more users to whom the distribution is being sent. If TOINTNET(*NONE) is specified, at least one user ID and address must be specified on the TOUSRID parameter. Up to 300 internet addresses can be specified.

The TOINTNET parameter is not allowed when the TYPE keyword is *MSG.

*NONE
> No internet address is specified. This is a single value.

character-value
> Specify the internet address of a person or organization to whom the distribution is being sent. Up to 253 characters can be specified for each address.

The possible Recipient Type values are:

*PRI   The internet address is the primary recipient of the distribution.

*CC   The internet address will receive a copy of the distribution sent to the primary recipient.

*BCC   The internet address will receive a copy of the distribution. However, this copy recipient is not identified on the distribution as a receiver on the distribution.

Description (DSTD)

Specifies the description of the distribution. A maximum of 44 characters can be specified. This parameter is required and can not start with a blank character or be all blank characters.

Message (MSG)

Specifies whether a short message is sent with the distribution.

*NONE
> No message is sent.

*DSTIDMSG
> The message in the distribution document specified on the **Distribution identifier** prompt (DSTID parameter) is sent with the distribution.

message-text
> Specify the message (256 characters maximum) to send to the users.

Message (LONGMSG)

Specify the text to send to the recipients as an final-form text document (FFTDCA).

*NONE
> No text is sent. This is the default and required when TYPE is not *LMSG

long-message-text
> Specify the text (5000 characters maximum) to send to the recipients. The text will be put into an final-form text document (FFTDCA) that will be sent as a note.
>
> The text must be formatted by using the following controls within the text.
> - :/N - Will cause a new line (Carrier return).
> - :/P - Will cause a new paragraph. New line plus a blank line. (Carrier return and Required carrier return)
>
> The document is setup with 10 characters to the inch, the font ID is 11, the left margin at position 1, and the right margin at position 75. If :/N is not used to start a new line, the text will continue past the right margin and will not be viewable by some E-mail clients. Blanks are not removed from the text, but left in place for indentation and alignment. No extra text (ie. Recipient List, Date/Time, Subject, or Sender) is put into the document.
>
> Technical note to programmers: Any of the FFTDCA controls (in hexadecimal) can be inserted in the text to provide more control of the format if the controls above are not enough.

Confirmation of delivery (CFMDEL)

Specifies whether the sender receives a confirmation of delivery notification when each receiver gets the distribution. Even though confirmation of delivery is not requested, the sender is still informed when the distribution is not delivered because of an user ID is not valid, a system failure, or a routing failure. If confirmation of delivery is requested, the sender is informed when the receiver receives, deletes, or runs

another command against the distribution. To get this information, the sender must use the Query Distribution (QRYDST) command with *OUT specified on the **Incoming or outgoing** prompt (OPTION parameter).

*NO   Confirmation of delivery is not requested.

*YES   Confirmation of delivery is requested.

Sensitivity (SENSITIV)

Specifies the level of sensitivity defined by the X.400 standard. The four levels include normal, personal, private and company confidential. Private distributions cannot be viewed by a user working on behalf of another user.

*NONE
   The distribution has no sensitivity restrictions.

*PERSONAL
   The distribution is sent to the recipient as an individual.

*PRIVATE
   The distribution contains information that should be accessed only by the recipient.

*CONFIDENTIAL
   The distribution contains information that should be handled according to company procedures.

Personal (PERSONAL)

Specifies whether the document distribution is private or not. This parameter is replaced by SENSITIV but the PERSONAL parameter can still be used. However, because this parameter may be removed in a later release, whenever possible use the SENSITIV parameter.

If PERSONAL(*YES) is used, the SENSITIV parameter must be omitted or it must be with the value SENSITIV(*NONE). If the command is prompted without this parameter specified, this parameter is not displayed.

*NO   Only the owner and users that have authorization to the distribution document can get access to distributions that are not sensitive. Users authorized to work on behalf of other users who have access to the distribution can access documents that are not sensitive. This value will map to SENSITIV(*NONE).

*YES   Only the owner can get access to private distribution documents. Users authorized to work on behalf of other users who have access to the distribution document cannot get access to the distribution. This value will map to SENSITIV(*PRIVATE).

Content importance (IMPORTANCE)

Specify low, normal or high importance. This is an indication to the recipient of the content importance of the distribution. It is different from priority which relates to the speed with which the distribution was sent.

*NORMAL
   A distribution of normal importance.

*HIGH

A very important distribution.

*LOW A low importance distribution.

Priority (PTY)

Specifies whether the distribution is sent using low priority, normal priority or high priority. For distributions to remote receivers, the priority determines which 'SNADS Next System Queue' is used. Normal priority distributions use the normal next system queue. Low priority distributions use the normal next system queue with the lowest priority available within that queue. High priority uses the priority next system queue. The difference between low, high and normal priority depends on information, specified by the user, on the Configuration Distribution Services (CFGDSTSRV) command. The handling of priority distributions by other office system nodes can vary, but generally the high priority distributions take the faster path when there is a choice of paths. For distribution to local receivers, the priority determines whether a message is sent to the receiver's message queue to notify the receiver of the distribution. No message is sent for normal distributions.

*NORMAL

Normal priority is used.

*HIGH

High priority is used.

*LOW Low priority is used.

User identifier (USRID)

Specifies which user ID and user ID address should be associated with the request.

*CURRENT

You are performing the request for yourself.

user-ID

Specify another user's user ID or your user ID. You must have been given permission to work on behalf of another user or have *ALLOBJ authority.

user-ID-address

Specify another user's address or your address. You must have been given permission to work on behalf of another user or have *ALLOBJ authority.

Document file (DOCFILE)

Specifies the names of the database file and the library that contains the document data. The database file is a user-defined file or the output file specified in either the Receive Distribution (RCVDST) command or the Retrieve Document (RTVDOC) command. If an output file is specified, only the data portion of the document data record is read from the output file. The prefix is removed from the document data record.

The possible library values are:

*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
    The current library for the job is used to locate the database file. If no library is specified as the library for the job, QGPL is used.

library-name
    Specify the library where the database file is located.

data-base-file-name
    Specify the name of the database file that contains the document data.

Document member (DOCMBR)

Specifies the document database file member that is used.

*FIRST
    The first member created in the database file is used.

member-name
    Specify the name of the database file member that is used.

Document type (DOCTYPE)

Specifies the type of document being used. This identifier is used by the system to determine whether the data stream can be handled properly.

*DFT   The system creates the proper document type identifier based on the source of the data.

*FFT   The document is in Final Form Text. This type of document is intended to be viewed and printed, but not edited, by the receiver.

*RFT   The document is in Revisable Form Text. This type of document can be viewed, printed, and edited by the receiver.

document-type-number
    Specify a document type identifier value ranging from 2 through 65,535. The numbers from 2 through 32,767 are controlled by registering them with the IBM Document Interchange Architecture and are used for IBM-defined document types. The numbers ranging from 32,768 through 65,535 are not registered with IBM and can be used for non-IBM-defined document types. The meaning of these document types must be determined by defining the value of the system code on the **System code** prompt (SYSCOD parameter).

Send format (SNDFMT)

Allows the user to specify the format of the document being sent.

*NOCHG
    The document is sent in the current format.

*NOTE
    The document is sent in a final form text document content architecture (FFTDCA) data stream as a note.

*FINALFORM
    The document is sent in FFTDCA.

System code (SYSCOD)

Specifies the text used with the value specified on the **Document type** prompt (DOCTYPE parameter) to help uniquely identify the type of document being used. The receiver of the data stream determines the document data stream and processing requirements to edit, view, print, or change the document.

*DFT The system supplies a default system code. If the value specified on the **Document type** prompt (DOCTYPE parameter) is a number ranging from 2 through 32,767, the default is 'IBM AS/400 CL' and is retrieved from message CPX9026. If the value specified on the **Document type** prompt (DOCTYPE parameter) is in the range from 32,768 through 65,535, a system code must be specified.

system-code
> Specify the text that uniquely identifies the type of document being sent. A maximum of 13 characters can be specified.

Document character identifier (DOCCHRID)

Specifies the character identifier (graphic character set and code page) for the document data being used. The character identifier is related to the display device used to create the document data.

*SYSVAL
> The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

*DEVD
> The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

graphic-character-set code-page
> Specify the graphic character set and code page values used to create the data being distributed.

Note: Both parts can be up to 5 digits in length.

Distribution identifier (DSTID)

Specifies the unique distribution identifier of the distribution. The distribution identifier is assigned by the system when the distribution is originated. Distribution identifiers can be found by using the Query Distribution (QRYDST) command. Identifiers are also returned from the Send Distribution (SNDDST) command.

*NONE
> No distribution identifier is used.

distribution-id
> Specify the 3-part distribution identifier which is composed of the second part of the sender's user ID (padded on the right to 8 characters), the first part of the sender's user ID (padded on the right to 8 characters), and a 4-digit zoned sequence number with leading zeros. For example, 'NEWYORK SMITH 0204'. This parameter is required when *DSTID is specified on the **Information to be sent** prompt (TYPE parameter).

Distribution ID extension (DSTIDEXN)

Specifies the extension of the distribution identifier (if any) specified on the **Distribution identifier** prompt (DSTID parameter). This extension uniquely identifies duplicate distributions. This 2-digit extension has a value ranging from 00 through 99 that uniquely identifies duplicate distributions. For incoming distributions, this extension ranges from 01 through 99. For confirmation of delivery distributions, this extension must be 00.

*NONE
> There is no duplicate distribution. *NONE is equivalent to an extension of 01.

distribution-id-extension
> Specify the extension associated with the distribution. This is used to uniquely identify duplicate distributions.

Document (DOC)

Specifies the name of the document being sent.

document-name
> Specify the user-assigned name (10 characters maximum) of the document to be sent.

*DOCID
> The document being sent is identified by the library-assigned document name.

In folder (FLR)

Specifies the name of the folder that contains the document being sent. This is the user-assigned name given to the folder when it is created. If document name is specified on the **Document** prompt (DOC parameter), then *DOCID must not be specified.

*NONE
> The document is not located in a folder.

folder-name
> Specify the name of the folder that contains the document being sent. A folder name can consist of a series of folder names if the document being sent is located in a folder contained within another folder or folders.

Document identifier (DOCID)

Specifies the library-assigned name of the document. This is the name assigned to the document by the system when it was created. Documents filed outside the local system have only library-assigned document names. The library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command or by the message returned from the File Document (FILDOC) command.

Library-assigned document names are 24 characters in length with the following format:

```
YYYYMMDDHHMNSSHSSNSNSNSN
```

where:

```
     YYYY = year
       MM = month
       DD = day
       HH = hour
       MN = minute
       SS = second
       HS = hundredths of a second
 SNSNSNSN = system name
```

*NONE

No library-assigned document name is required when the document is identified on the **Document** prompt (DOC parameter).

library-assigned-document-name

Specify the library-assigned name of the document being sent.

Profile file (IDPFILE)

Specifies where the document profile information is located. If you specify this parameter, the remaining parameters after the **Profile member** prompt (IDPMBR parameter) are ignored, except the **Command character identifier** prompt (CMDCHRID parameter) and the **Document character identifier** prompt (DOCCHRID parameter).

*NONE

The interchange document profile (IDP) is supplied by other parameters on this command. There is no database file containing the IDP information. If *NONE is specified, the **Profile member** prompt (IDPMBR parameter) is ignored.

*DSTIDIDP

The IDP information associated with the distribution document is used. The **Profile member** prompt (IDPMBR parameter) is ignored. This is valid only when TYPE (*DSTID) is specified.

*DOCFILE

The database file specified for the document also contains the profile information. If *DOCFILE is specified, the **Document file** prompt (DOCFILE parameter) and **Document member** prompt (DOCMBR parameter) are used for the document profile information.

data-base-file-name

Specify the name of the database file that contains the IDP. The document profile database file can be a user-defined file or the output file specified on the Receive Distribution (RCVDST) or Retrieve Document (RTVDOC) commands. If you specify a user-defined file, it must have the same format as the output file produced by RCVDST or RTVDOC. If an output file is specified, only the data portion of the document profile record is read from the output file. The prefix is removed from the document profile record.

The possible library values are:

*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB

The current library for the job is used to locate the database file. If no current library is specified as the library for the job, QGPL is used.

library-name

Specify the library where the database file is located.

Profile member (IDPMBR)

Specifies the interchange document file member name being used. This parameter is used only when a database file name is also specified on the **Profile file** prompt (IDPFILE parameter).

*FIRST
> The first member created in the database file is used.

member-name
> Specify the name of the database file member being used.

Document description (DOCD)

Specifies a description for the document being distributed. This information is in the Document Interchange Architecture IDP document name field.

*DFT   The system generates a document description. For database files, the default format is library-name/file-name/member-name. For a description of a hard copy document, the default is the distribution description. For a description of a distribution document, the default is the document description associated with the distribution.

*DSTD
> The distribution description specified on the **Description** prompt (DSTD parameter) is used for the document description.

document-description
> Specify the description of the document. A maximum of 44 characters can be specified.

Author (AUTHOR)

Specifies the author or authors of the document.

You can enter multiple values for this parameter.

*NONE
> No author is identified for the document.

*USRID
> The user ID and address specified on the USRID parameter **User identifier** prompt(USRID parameter) is used as the author's name.

document-author-name
> Specify the name of the author or authors. A maximum of 50 authors can be specified.

Document class (DOCCLS)

Specifies the class associated with this document, such as MEMO, FORM, or SHEET.

*NONE
> No class is assigned to the document.

document-class
> Specify the document class. A maximum of 16 characters can be specified.

Keyword (KWD)

Specifies the keywords that describe the document.

You can enter multiple values for this parameter.

*NONE
> No keywords are defined for this document.

document-keyword
> Specify the keywords to describe the document. A maximum of 50 keywords can be specified. Each keyword can have a maximum of 60 characters.

Subject (SUBJECT)

Specifies the subject or subjects of the document.

You can enter multiple values for this parameter.

*NONE
> No subject is defined for the document.

*DOCD
> The document description is used as the subject for the document.

document-subject
> Specify the subject or subjects of the document. A maximum of 50 subjects can be specified and each subject can have a maximum of 60 characters of text.

Document date (DOCDATE)

Specifies any date the user needs to assign to the document.

*NONE
> No date is assigned to the document.

*CURRENT
> The system assigns the current system date to the document.

document-date
> Specify the document date. The date must be specified in the job date format.

File cabinet location (FILCAB)

Specifies the location of the document. This parameter is intended to describe the location of printed documents. The interchange document profile (IDP) that refers to the printed document is distributed. This parameter is required if *IDP is also specified on the **Information to be sent** prompt (TYPE parameter) and *NONE is specified on the **Profile file** prompt (IDPFILE parameter).

*NONE
No filing cabinet reference is defined for this document.

filing-cabinet-reference
Specify the text that describes where the printed document is located. A maximum of 60 characters can be specified.

Top

Copy list (CPYLST)

Specifies the names of the users that receive this document.

You can enter multiple values for this parameter.

*NONE
No copy list is included for this document.

recipient-list
Specify the names of the users that receive the document. A maximum of 50 names can be specified. Each name can have a maximum of 60 characters.

Top

Expiration date (EXPDATE)

Specifies the date on which the document is no longer needed.

*NONE
No document expiration date is specified.

expiration-date
Specify the document expiration date. The date must be specified in the job date format.

Top

Creation date (CRTDATE)

Specifies the date the document was created.

*NONE
No document creation date is specified.

*CURRENT
The current system date is used as the date the document was created.

create-date
Specify the document creation date. The date must be specified in the job date format.

Top

Reference (REFERENCE)

Specifies a reference associated with the document.

*NONE

> No reference field is included for this document distribution.

reference

> Specify text that describes the reference associated with the document. A maximum of 60 characters can be used.

Action due date (ACTDATE)

Specifies the due date for the requested action. If no action due date is specified and a reply request date is specified the action due date is set from the reply request date.

*NONE

> No action due date is specified.

*CURRENT

> The current date is used.

action-due-date

> Specify the value used as the action due date. The date must be specified in the format specified by the system value QDATFMT.

Reply requested (RPYDATE)

Specifies the date and time a reply to the distribution is requested. This applies only to primary recipients.

The possible Reply by date values are:

*NONE

> No reply is required.

*CURRENT

> The current date is used. The time defaults to *ENDOFDAY and is set to 23:59:59.

*ANY   A reply is requested but no date and time are specified.

reply-by-date

> Specify the value used as the reply requested date. The date must be specified in the format specified by the system value QDATFMT.

The possible Reply by time values are:

*ENDOFDAY

> A reply is requested by the end of the specified date. The time is set to 23:59:59.

reply-by-time

> Specify the value used as the reply requested time.
>
> The time is specified in 24-hour format and can be specified with or without a time separator.

Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh=hours, mm=minutes, and ss=seconds. Valid values for hh range from 00 to 23. Valid values for mm and ss range from 00 to 59.

With a time separator, specify a string of 5 or 8 digits (hours and minutes, or hours, minutes, and seconds). The time separator specified by the system value QTIMSEM is used to separate the hours, minutes, and seconds. If the command is entered from the command line, the string must be entered in apostrophes. If a time separator other than the separator used for your job is used, the command fails. Valid values for hh range from 00 to 23. Valid values for mm and ss range from 00 to 59.

Document status (STATUS)

Specifies the user-defined status of the document. Examples of status are: In Process, Pending Approval, or Retired.

*NONE
  No status is included in this document.

status-of-document
  Specify text that describes the status of the document. A maximum of 20 characters can be specified.

Completion date (CMPDATE)

Specifies the date when the requested action is completed.

*NONE
  No completion date is included.

*CURRENT
  The current system date is used as the completion date.

date-complete
  Specify the action completion date. The date must be specified in the job date format.

Project (PROJECT)

Specifies the project associated with the document.

*NONE
  No project field information is included in this document.

project
  Specify text that describes the project of the document. A maximum of 10 characters can be specified.

Language ID (DOCLANGID)

Specifies the language identifier to be placed in this document's interchange document profile (IDP).

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified, or if *MSG or *DOC is specified on the **Information to be sent** prompt (TYPE parameter).

*JOB   The language identifier specified for the job in which this command is entered is used.

language-identifier
>   Specify a language identifier. Press the F4 key from the **Language ID** prompt (DOCLANGID parameter) to see a list of valid identifiers.

Top

Country or region ID (DOCCNTRYID)

Specifies the country or region identifier to be placed in this document's interchange document profile (IDP).

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified, or if *MSG or *DOC is specified on the **Information to be sent** prompt (TYPE parameter).

*JOB   The country or region identifier specified for the job in which this command is entered is used.

country-or-region-identifier
>   Specify a country or region identifier. Press the F4 key from the **Country or region ID** prompt (DOCCNTRYID parameter) to see a list of valid identifiers.

Top

Allow alternate recipient (ALWALTRCP)

Specify if the distribution can be delivered to an alternate recipient, determined by the receiving system. This function is used only by X.400 network systems. This function provides the ability to direct certain mail to a specific user.

The default of this field is *YES. Because the user might change the sensitivity field to private but might not change the alternate recipient field to no, The user might send a private item to an alternate recipient. No error message is generated for this condition.

*YES   Specifies the distribution can be delivered to an alternate recipient.

*NO   Specifies the distribution cannot be delivered to an alternate recipient.

Top

Disclose recipient (DISCLOSE)

Specifies whether or not each recipient gets a list of the other recipients.

*YES   Disclose recipients.

*NO   Do not disclose recipients.

Top

Allow X.400 conversion (ALWX400CNV)

Specifies whether an X.400 conversion is allowed on the distribution being submitted. In certain circumstances, the end user knows the capabilities of the recipient and may indicate that the distribution is not to be converted. The system administrator can define whether or not conversion is to take place. This field allows the user to override that setting on a message by message basis. For example, if the gateway allows conversion, the user could specify that conversion is to be prohibited. This will take precedence over the gateway setting. The distribution would then not be converted.

Note: This field is valid only for X.400 support. This field will not affect AS/400 data stream transformations, such as RFT to FFTDCA.

*YES   Conversion may be performed on the distribution by the receiving system.

*NO   No conversion is permitted on the distribution by the receiving system.

Authorizing user (AUTUSR)

Specifies the user ID and address of the user that authorized the content of this distribution. The authorizing user will receive a copy of the distribution.

*NONE
    No authorizing user.

The possible User ID value is:

user-ID
    Specify the user ID of the user from whom the distribution is authorized.

The possible User Address value is:

user-address
    Specify the user address of the user from whom the distribution is authorized.

Distribution expiry indicator (DSTEXPDATE)

Specifies the date and time on which the distribution is no longer needed in the mail log.

The possible Distribution Expiration Date values are:

*NONE
    The distribution has no expiration date.

dist-expiration-date
    Specify the value to use as the expiration date for the distribution. The date must be in the format specified by the system value QDATFMT.

The possible Distribution Expiration Time values are:

*ENDOFDAY
    An expiration time is requested by the end of the specified date. The time is set to 23:59:59.

dist-expiration-time
    Specify the value as the expiration time.

    The time is specified in a 24-hour format and can be specified with or without a time separator.

Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh=hours, mm=minutes, and ss=seconds. Valid values for hh range from 00 to 23. Valid values for mm and ss range from 00 to 59.

With a time separator, specify a string of 5 or 8 digits (hours and minutes, or hours, minutes, and seconds). The time separator specified by the system value QTIMSEM is used to separate the hours, minutes, and seconds. If the command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator used for your job is used, the command fails. Valid values for hh range from 00 to 23. Valid values for mm and ss range from 00 to 59.

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values. The character identifier is related to the display device used to enter the command.

In some cases, the data is translated to a code page and character set that is interchangeable with other IBM OfficeVision/400 products. The interchangeable character set and code page is '697 500', except for the **User identifier** prompt (USRID parameter), **Recipient** prompt (TOUSRID parameter), and **Distribution identifier** prompt (DSTID parameter), for which it is '930 500'. In other cases, the code page and character set are attached to the field and sent with the field to allow the receiving terminal to correctly print and display the field.

The following parameters are translated:
- **Recipient** (TOUSRID)
- **Internet Recipient** (TOINTNET)
- **User identifier** (USRID)
- **Distribution identifier** (DSTID)
- **Document system code** (SYSCOD)
- **Message** (MSG)
- **Description** (DSTD)

The code page and character set is attached to the following parameters:
- **Long Message** (LONGMSG)
- **Document description** (DOCD)
- **Author** (AUTHOR)
- **Document class** (DOCCLS)
- **Keyword** (KWD)
- **Subject** (SUBJECT)
- **File cabinet location** (FILCAB)
- **Copy list** (CPYLST)
- **Reference** (REFERENCE)
- **Document status** (STATUS)
- **Project** (PROJECT)

Single values

*SYSVAL

The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

*DEVD

The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

Element 2: Code page

1-32767

Specify the code page to use.

Top

Examples

Example 1: Sending a Distribution on Behalf of Another User

```
SNDDST   TYPE(*FILE)  TOUSRID((JACKSON RCH38DB))
         DOCTYPE(20000)  SYSCOD(BRANDX)
         DOCFILE(DEPT46ELIB/XTEXT)  DOCMBR(GOLD1IPFS)
         PTY(*HIGH)  USRID(JACOBSON RCH38NBS)
         DSTD('IPFS FOR GOLD1 PROJECT')  CFMDEL(*YES)
         MSG('Update section 1.2.4. Return for final printing')
```

This command sends a distribution that is being sent by someone (such as a secretary) who is authorized to work on behalf of JACOBSON. The document being sent is a BRANDX text document that is sent to another user who also has the BRANDX text processor.

Example 2: Sending a Mail Log Entry

```
SNDDST   TYPE(*DSTID)  DSTID('NEWYORK SMITH 0204')
         DSTIDEXN(02)  TOUSRID((JACKSON RCH38DB))
         MSG(*DSTIDMSG)  CFMDEL(*YES)
```

This command sends a mail log entry that is identified by the distribution document name **NEWYORK SMITH 0204** that is distributed to user JACKSON at address RCH38DB. The message in the distribution document is distributed with the distribution.

Top

Error messages

*ESCAPE Messages

CPF8A87

Document name &2 not correct.

CPF8A97

Folder name &1 not correct.

CPF89AA

*FINALFORM for send format (SNDFMT) valid only for *DOC for type (TYPE).

CPF89AB

Specify *IDP or *MSG for type (TYPE) only with *NOCHG for send format (SNDFMT).

CPF900B

User ID and address &1 &2 not in System Distribution Directory.

CPF900C

Sign on and verify of user failed.

CPF901A

Send distribution request failed.

CPF903D

Incorrect document identifier specified.

CPF905C

Error occurred trying to find a translation table.

CPF9096

Cannot use CMDCHRID(*DEVD), DOCCHRID(*DEVD) in batch job.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

Error occurred while closing file &1 in library &2.

Top

Send Distribution Queue (SNDDSTQ)

The Send Distribution Queue (SNDDSTQ) command is used:

- To send a distribution queue's entries when the distribution queue is configured to be manually started but no operator is available.
- To override any distribution queue scheduling attributes and begin sending a queue's entries immediately.
- To restart a SNADS sender job that failed abnormally.

The SNDDSTQ command is primarily intended for use in a batch CL program. The SNDDSTQ command enables the same functions as option 2 (Send distribution queue) on the Work with Distribution Queue (WRKDSTQ) command main list panel. The SNDDSTQ command allows the functions to be started from a batch job instead of interactively.

Distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

- This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
- Messages that report errors about distribution queues may display or print different characters than you entered for the distribution queue name because of internal system transformations. Similarly (depending on the language used for the work station), the internal value for a distribution queue name may differ from the characters shown for the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the **Distribution queue** prompt (DSTQ parameter) does not match the rules for an internal distribution queue value or if it does not match the internal value for any defined distribution queue (ignoring case differences).

Parameters

Keyword	Description	Choices	Notes
DSTQ	Distribution queue	*Character value*	Required, Positional 1
PTY	Priority	*NORMAL, *HIGH	Required, Positional 2

Distribution queue (DSTQ)

Specifies the name of the distribution queue that is sent. The queue must have been previously configured using the Configure Distribution Services (CFGDSTSRV) command or the Add Distribution Queue (ADDDSTQ) command.

This is a required parameter.

Priority (PTY)

Specifies whether the normal priority or high priority portion of the specified queue is sent.

The possible values are:

*NORMAL
>   The normal priority queue is for those distributions with a service level of data low.

*HIGH
>   The high priority queue is for those distributions with a service level of fast, status, or data high.

>   **Note:** This value is not valid for a SystemView distribution services (SVDS) type of distribution queue.

This is a required parameter.

Examples

Example 1: Sending Distributions with Normal Priority

```
SNDDSTQ   DSTQ(CHICAGO)   PTY(*NORMAL)
```

This command sends distributions from the normal priority portion of the CHICAGO distribution queue.

Example 2: Sending Distributions with High Priority

```
SNDDSTQ   DSTQ(ATLANTA)   PTY(*HIGH)
```

This command sends distributions from the high priority portion of the ATLANTA distribution queue.

Error messages

*ESCAPE Messages

CPF8802
>   Distribution queue &1 was not found.

CPF8805
>   Special value for System name/Group not permitted or not used correctly.

CPF8806
>   Value &1 not valid for system name or system group.

CPF881C
>   High priority queue not allowed for *SVDS distribution queue &1

CPF8812

　　　　Error occurred while processing distribution queues.

CPF8816

　　　　QSNADS communications subsystem is not active.

CPF8817

　　　　Distribution queue is held.

CPF9845

　　　　Error occurred while opening file &1.

CPF9846

　　　　Error while processing file &1 in library &2.

CPF9847

　　　　Error occurred while closing file &1 in library &2.

Top

Send File (SNDF)

Threadsafe: No

The Send File (SNDF) command is used by a CL procedure to send a record to a display device that is being used by an interactive user. The device can be any display station, including the console. The command sends the data from the program's CL variables to the display's device file in the specified record format. These variables were automatically declared in the program (one for each field in the record format) when the CL source program was compiled and a Declare File (DCLF) command was processed as part of the source.

Of the record formats specified in the DCLF command, only one can be specified in each SNDF command. If the device file has not been opened, it is opened by this command. The file and record format specified in this command can be overridden by an Override with Display File (OVRDSPF) command if it is entered before the file is opened. However, care should be taken that the fields in the overriding record format correspond to the CL variables declared in the program.

Restrictions:
- This command is valid only within a CL procedure.
- This command is valid only for display files.
- This command cannot be used with database files.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Display device	*Name*, ***FILE**	Optional, Positional 1
RCDFMT	Record format	*Name*, ***FILE**	Optional, Positional 2
OPNID	Open file identifier	*Simple name*, ***NONE**	Optional

Top

Display device (DEV)

Specifies the name of the display device to which the data in the CL variables for the specified record format is to be sent.

*FILE   The program's data is to be sent to the device associated with the device file that was declared in the FILE parameter of the DCLF command. If more than one device name is specified in the device file, *FILE cannot be specified.

name   Specify the name of the device or the name of the CL variable that contains the name of the device to which the program's data is to be sent.

Record format (RCDFMT)

Specifies the name of the record format that is to be used to send data to the file. The format contains all the fields in the record. This parameter must be coded with a record format name if there is more than one record format name in the device file; *FILE cannot be coded if there is more than one. If the record format contains the INVITE DDS keyword (optioned on), the SNDF functions as if SNDRCVF WAIT(*NO) had been coded.

*FILE There is only one record format in the device file; that is the format in which the program's data is to be sent to the file.

name Specify the name of the record format in which the program's data is to be sent to the file. A CL variable cannot be used to specify the record format name.

Open file identifier (OPNID)

Specifies the open file identifier that was declared on a preceding Declare File (DCLF) command in the same CL procedure. A CL variable cannot be specified for this parameter value.

*NONE
> No open file identifier is provided. This command will use the file associated with the DCLF command that had *NONE specified for the OPNID parameter. Only one file can be declared in a CL procedure with *NONE as the open file identifier.

simple-name
> Specify a name that matches the OPNID parameter value on a preceding DCLF command in the same CL procedure.

Examples

Example 1: Using Display File with One Record Format

```
DCLF    FILE(MENU1)
 :
SNDF
```

The record format in the device file MENU1 is sent to the device specified in the file. There is only one record format in the file.

Example 2: Using Display File with Multiple Record Formats

```
DCLF    FILE(SCREEN1)  RCDFMT(REC1 REC2)
 :
SNDF    DEV(DISP3)  RCDFMT(REC1)
```

The device file named SCREEN1 causes the display station named DISP3 to display the data sent by the CL procedure. The data is shown in the format specified by the REC1 record format.

Example 3: Using Open File Identifier

```
DCLF    FILE(SCREEN1)  RCDFMT(REC1 REC2)  OPNID(OUTDSP1)
DCLF    FILE(SCREEN2)  RCDFMT(REC3 REC4)  OPNID(OUTDSP2)
 :
SNDF    DEV(*FILE)  RCDFMT(REC2)  OPNID(OUTDSP1)
```

The device file named SCREEN1 is used to send data to the display device named in the same device file. The data is presented to the user in the format specified by record format REC2. The SNDF command is associated with device file SCREEN1 because the open file identifier specified on the SNDF command matches the open file identifier specified on the DCLF command for display file SCREEN1.

Error messages

*ESCAPE Messages

CPF0859
>   File override caused I/O buffer size to be exceeded.

CPF0861
>   File &1 in library &2 is not a display file.

CPF0864
>   End of file detected for file &1 in &2.

CPF0883
>   *FILE not valid in DEV parameter for file &1.

CPF0887
>   Data available from previous input request.

CPF4101
>   File &2 in library &3 not found or inline data file missing.

CPF5068
>   Program device &4 not found in file &2 in library &3.

CPF5070
>   File &2 in library &3 has no program devices acquired.

Send Journal Entry (SNDJRNE)

The Send Journal Entry (SNDJRNE) command is used to write a single journal entry to a specific journal. The entry can contain any information. The user may assign an entry type to the journal entry and may also associate the journal entry with a specified journaled object.

If the journal currently has a state of *STANDBY, then the journal entry will not be deposited unless OVRSTATE(*STANDBY) is specified.

The journal code for the entry is U, which indicates a user-specified journal entry.

Note: The Send Journal Entry (QJOSJRNE) Application Programming Interface (API) can also be used to write a user-specified journal entry to a specific journal. Using this API may improve performance and can provide additional function that is not available with this command. For more information, see the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Restrictions:

- If an object other than a file is specified, it must be currently journaled to the specified journal.

  If a file object is specified, it must either be currently journaled to the specified journal or it must have been last journaled to the specified journal.

- The specified journal cannot be a remote journal.
- The specified journal cannot have a journal state of *INACTIVE.

Parameters

Keyword	Description	Choices	Notes
JRN	Journal	*Qualified object name*	Required, Positional 1
	Qualifier 1: Journal	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TYPE	Journal entry type	*Character value*, <u>**00**</u>	Optional, Positional 2
ENTDTA	Entry data	*Character value*, ***BLANK**	Optional, Positional 3
FILE	Journaled physical file	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Physical file	*Qualified object name*	
	Qualifier 1: Physical file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Member	*Name*, ***FIRST**, *NONE	

Keyword	Description	Choices	Notes
OBJ	Object	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Object type	***FILE**, *DTAARA, *DTAQ	
	Element 3: Member	*Name*, ***FIRST**, *NONE	
OBJPATH	Object	*Path name*, ***NONE**	Optional
OBJFID	File identifier	*Hexadecimal value*, ***NONE**	Optional
FORCE	Force journal entry	***NO**, *YES	Optional
OVRSTATE	Override journal state	***NONE**, *STANDBY	Optional

<div align="right">Top</div>

Journal (JRN)

Specifies the journal to contain the new journal entry.

This is a required parameter.

Qualifier 1: Journal

journal-name
> Specify the name of the journal to contain the new journal entry.

Qualifier 2: Library

*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
> The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

library-name
> Specify the name of the library to be searched.

<div align="right">Top</div>

Journal entry type (TYPE)

Specifies the journal entry type of this journal entry.

<u>00</u>   The journal entry type is a '00' (hex F0F0).

entry-type
> Specify a 2-character value or hex value used for the journal entry type. This value must be greater than or equal to hex C000.
>
> If a hexadecimal value is specified that does not represent characters, that value is not shown on the DSPJRN display or on the printout.

<div align="right">Top</div>

Entry data (ENTDTA)

Specifies the user-specified data that is placed in the variable portion of the journal entry.

*BLANK
>No user-specified data is placed in the journal entry.

'entry specific-data'
>Specify up to 3000 characters, enclosed in apostrophes.

Journaled physical file (FILE)

Specifies the database physical file and member with which this entry is associated.

If this parameter is specified, the parameters OBJ, OBJPATH or OBJFID cannot also be specified.

Single values

*NONE
>There is no associated physical file for this entry.

Element 1: Physical file

Qualifier 1: Physical file

file-name
>Specify the name of the physical file with which this entry is associated.

Qualifier 2: Library

*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
>The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

library-name
>Specify the name of the library to be searched.

Element 2: Member

*FIRST
>The entry is associated with the first member in the file.

*NONE
>The entry is associated with the file, not with any member of the file.

member-name
>Specify the name of the physical file member with which this entry is associated.

Object (OBJ)

Specifies the object with which this entry is associated.

If this parameter is specified, the parameters FILE, OBJPATH or OBJFID cannot also be specified.

Single values

*NONE
> There is no associated object for this entry.

Element 1: Object

Qualifier 1: Object

object-name
> Specify the name of the object with which this entry is associated.

Qualifier 2: Library

*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
> The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

library-name
> Specify the name of the library to be searched.

Element 2: Object type

*FILE The entry is associated with a database file or database file member.

*DTAARA
> The entry is associated with a data area.

*DTAQ
> The entry is associated with a data queue.

Element 3: Member

*FIRST
> The entry is associated with the first member in the file.

*NONE
> The entry is associated with the file, not with any member of the file.

member-name
> Specify the name of the physical file member with which this entry is associated.

> **Note:** If the specified object type is not *FILE, the member name value is ignored.

Top

Object (OBJPATH)

Specifies the path-qualified object name with which this entry is associated. Only objects whose path name identifies an object of type *STMF, *DIR or *SYMLNK that is in the "root" (/), QOpenSys, and user-defined file systems are supported.

If this parameter is specified, the parameters FILE, OBJ or OBJFID cannot also be specified.

*NONE
> There is no associated object for this entry.

path-name
> Specify the name of the object with which this entry is associated.

> A pattern can be specified in the last part of the path name. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes. Symbolic links within the path name will not be followed. If the path name begins with the tilde character, then the path is assumed to be relative to the appropriate home directory.

> Additional information about path name patterns is in the Integrated file system information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

File identifier (OBJFID)

Specifies the file-identifier (FID) with which this entry is associated. FIDs are a unique identifier associated with integrated file system related objects. This field is input in Hexadecimal format. Only objects whose FID identifies an object of type *STMF, *DIR or *SYMLNK that is in the "root" (/), QOpenSys, and user-defined file systems are supported.

If this parameter is specified, the parameters FILE, OBJ or OBJPATH cannot also be specified.

*NONE
> There is no associated object for this entry.

file-identifier
> Specify the FID for the object with which this entry is associated.

Force journal entry (FORCE)

Specifies whether the journal receiver is forced to auxiliary storage after the user entry is written to it.

*NO   The journal receiver is not forced to auxiliary storage.

*YES   The journal receiver is forced to auxiliary storage.

Override journal state (OVRSTATE)

Specifies whether the journal entry will be deposited, overriding the current state of the journal.

*NONE
> None of the journal state values are overridden. That is,
> - The journal entry is deposited if the journal state is *ACTIVE.
> - The journal entry is not deposited and an error is not sent if the journal state is *STANDBY.

*STANDBY
> The journal entry is deposited even if the journal state is *STANDBY.

Examples

Example 1: Forcing Journal Receivers to Auxiliary Storage

```
SNDJRNE   JRN(JRNLA)  TYPE(AB)  ENTDTA('PROGRAM COMPLETE')
          FILE(MYLIB/ORDERENT MBR1)  FORCE(*YES)
```

If the journal currently has a journal state of *ACTIVE, this command places a journal entry of type AB (hex C1C2) with the journal entry data 'PROGRAM COMPLETE' in the current journal receivers attached to journal JRNLA as found by using the library search list. The entry is associated with member MBR1 of file ORDERENT in library MYLIB. The journal receiver is forced to auxiliary storage after the entry has been placed in it.

Example 2: Sending a Journal Entry

```
SNDJRNE   JRN(JRNLA)  TYPE(x'C1F1')  OVRSTATE(*STANDBY)
```

If the journal currently has a journal state of *STANDBY or *ACTIVE, this command places a journal entry of type 'A1' (hex C1F1) with no journal entry data in the current journal receiver attached to journal JRNLA as found by using the library search list. The entry is not associated with any physical file member.

Error messages

<u>*ESCAPE Messages</u>

CPFA0D4

File system error occurred. Error number &1.

CPF7002

File &1 in library &2 not a physical file.

CPF7003

Entry not journaled to journal &1. Reason code &3.

CPF7004

Maximum number of objects journaled to journal &1.

CPF7007

Cannot allocate member &3 file &1 in &2.

CPF7035

Object &1 in &2 already known to journal.

CPF7037

Object not journaled to journal &3.

CPF70EF

Parameters cannot be used together.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9810

Library &1 not found.

CPF9812

File &1 in library &2 not found.

CPF9815

Member &5 file &2 in library &3 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

Top

Send Message (SNDMSG)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

The Send Message (SNDMSG) command is used by a display station user to send an immediate message from his display station to one or more message queues. (An immediate message is a message that is not predefined and is not stored in a message file.) The message can be sent to the system operator, to other display station users, to a user's message queue, all currently active users' message queues or to the system history log, QHST. The sender can require a reply from the message receiver. The primary users of this command are display station users and the system operator.

Restrictions:

1. You must have object operational (*OBJOPR) and add (*ADD) authorities for the message queue.
2. You must have use (*USE) authority for the specified message queues and *USE authority for the libraries in which they are located.
3. The SNDMSG command only allows a message of up to 512 characters of first-level message text to be sent.
4. This command can only send inquiry messages (specified by MSGTYPE(*INQ)) to one message queue or to two message queues if one of the queues is *HSTLOG.

Parameters

Keyword	Description	Choices	Notes
MSG	Message text	*Character value*	Required, Positional 1
TOUSR	To user profile	*Name*, *SYSOPR, *ALLACT, *REQUESTER	Optional, Positional 3
TOMSGQ	To message queue	Single values: *SYSOPR Other values (up to 50 repetitions): *Qualified object name*	Optional, Positional 2
	Qualifier 1: To message queue	*Name*, *HSTLOG	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MSGTYPE	Message type	***INFO**, *INQ	Optional, Positional 4
RPYMSGQ	Message queue to get reply	Single values: ***WRKSTN** Other values: *Qualified object name*	Optional
	Qualifier 1: Message queue to get reply	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
CCSID	Coded character set ID	1-65535, *HEX, ***JOB**	Optional

Message text (MSG)

Specifies the immediate message that is being sent. The text must be enclosed in apostrophes if it contains blanks or other special characters. A maximum of 512 characters can be specified.

Coded Character Set Identifier (CCSID) Considerations

The text supplied for the MSG parameter is assumed to be in the CCSID of the job running this command unless the coded character set identifier is supplied for the CCSID parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

This is a required parameter.

To user profile (TOUSR)

Specifies that the message is to be sent to the message queue specified in the user profile for the user named on this parameter. This parameter cannot be used if a value is specified for the **To message queue(TOMSGQ)** parameter.

Either this parameter or the **To message queue (TOMSGQ)** parameter is required.

user-profile-name
> Specify the user profile name of the user to whom the message is sent.

*SYSOPR
> The message is sent to the system operator message queue, QSYS/QSYSOPR. Any message sent to QSYSOPR automatically has a copy of the message sent to QHST.

*REQUESTER
> The message is sent to the user profile message queue for interactive jobs or to the system operator's message queue (QSYS/QSYSOPR) for batch jobs.

*ALLACT
> A copy of the message is sent to the user profile message queue of each user profile with an interactive job currently running. *ALLACT cannot be specified with inquiry messages.

To message queue (TOMSGQ)

Specifies one to fifty message queues to which an informational message is sent. For an inquiry message, one message queue may be specified or two message queues may be specified if one of the queues is *HSTLOG. This parameter cannot be used if a value is specified for the **To user profile (TOUSR)** parameter.

Either this parameter or the **To user profile (TOUSR)** parameter is required.

Single values

*SYSOPR
> The message is sent to the system operator message queue, QSYS/QSYSOPR. Any message sent to QSYSOPR automatically has a copy of the message sent to QHST.

Qualifier 1: To message queue

*HSTLOG
> The message is sent to the system history log message queue, QSYS/QHST. If *HSTLOG is specified more than once, only one message will be sent to QSYS/QHST. If *HSTLOG is specified with QSYSOPR only one message is sent to QSYS/QHST.

message-queue-name
> Specify the name of the message queue to which the message is to be sent.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
> The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name
> Specify the library where the message queue is located.

Message type (MSGTYPE)

Specifies the type of message to be sent. Only an informational or inquiry message can be specified.

*INFO
> An informational message is to sent.

*INQ   An inquiry message is to be sent. The message queue receiving the message can reply to it. Inquiry messages are sent to only one message queue at a time or a second queue can be specified for TOMSGQ if the value is *HSTLOG.

Message queue to get reply (RPYMSGQ)

Specifies, only if an inquiry message is sent, the message queue to which a reply is sent.

Single values

*WRKSTN
> The reply to the message is sent to the display station message queue associated with the sender's display station.
>
> **Note:** This value cannot be specified for batch jobs.

Qualifier 1: Message queue to get reply

message-queue-name
> Specify the name of the message queue to which a reply is sent. Only a user message queue, a display station message queue, or the system operator message queue can be specified.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
> The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name
>Specify the library where the message queue is located.

Coded character set ID (CCSID)

Specifies the coded character set identifier (CCSID) that the supplied message text is in. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

*JOB The message text is assumed to be in the CCSID of the job running this command.

*HEX The message text is not converted. CCSID 65535 is used.

coded-character-set-identifier
>Specify a valid CCSID in which you want your message text to be considered in. Valid values are between 1 and 65535. This command validates the CCSID.

Examples

Example 1: Sending Message to User Message Queue

```
SNDMSG   MSG('Do you want to update INV now?')  TOUSR(JONES)
         MSGTYPE(*INQ)  RPYMSGQ(SMITH)
```

This command sends a message to the user message queue JONES. When the message is answered, the reply will be sent to the message queue SMITH.

Example 2: Sending Message to System's History Log

```
SNDMSG   MSG('Errors on PAYROLL cost me 1 hour of run time.')
         TOMSGQ(QHST)
```

This command is used by the system operator to send an informational message to the system's history log, QHST, through the log's message queue, which has the same name.

Example 3: Sending Message to System Operator

```
SNDMSG   MSG('Please make 2 copies of file LABORSTAT.')
         TOMSGQ(QSYSOPR)
```

This command shows a typical use of the SNDMSG command by a display station user. The user is sending the message to the system operator.

Example 4: Sending Message that Requires a Reply

```
SNDMSG   MSG ('How long will the system be up today?')
         TOMSGQ(*SYSOPR)  MSGTYPE(*INQ)
```

This command sends an inquiry message to the system operator. The message requires a reply. The system operator displays the message by using the DSPMSG command and enters the reply on the display. The reply is then sent to the display station user's work station message queue. The display station user enters another DSPMSG command to display the reply.

Error messages

CPF2428

Message queue parameter is not valid.

CPF2433

Function not allowed for system log message queue &1.

CPF2469

Error occurred when sending message&1.

CPF247E

CCSID &1 is not valid.

CPF2488

Reply message queue *WRKSTN not valid for batch job.

CPF9830

Cannot assign library &1.

CPF9838

User profile storage limit exceeded.

Top

Send Network File (SNDNETF)

The Send Network File (SNDNETF) command sends a save file or a member of a physical database file to another user on the local system or on a remote system through the SNADS network. This command can be used to:

- Send data files to a user.
- Send source files to a user. Source sequence information is kept in the file sent.
- Send other object types stored in a save file to a user.

When the file arrives at its destination, a notification message is sent to both the recipient and sender of the file.

When a source physical file is sent, the source sequence number and change date in positions 1 through 12 of the record are sent with the file. These are kept if the file is received into a source physical file, and are truncated if the file is received into a nonsource physical file. When a file that was originally a nonsource physical file is received into a source physical file, the source sequence numbers are created and placed in front of the records.

Note: Save files created on the AS/400 system cannot be distributed to System/38. However, save files created on System/38 **can** be distributed to the AS/400 system.

This command does not perform any CCSID translation on the contents of the file. However, the user ID and address of both the recipient and the originator are translated from the current job CCSID to the multinational character set 697/500.

Restrictions:

1. The user must be enrolled in the system distribution directory.
2. The maximum size of a file that can be sent using the SNDNETF command is approximately 2 billion bytes.

Parameters

Keyword	Description	Choices	Notes
FILE	File	*Qualified object name*	Required, Positional 1
	Qualifier 1: File	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TOUSRID	User ID	Values (up to 50 repetitions): *Element list*	Required, Positional 2
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	
MBR	Member	*Name*, ***FIRST**	Optional, Positional 3
TOTYPE	To file type	***FROMFILE**, *DATA	Optional

Keyword	Description	Choices	Notes
CLASS	VM/MVS class	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Optional
FORMAT	Send format	*V, *F, *U	Optional
PTY	Send priority	*NORMAL, *HIGH	Optional

File (FILE)

Specifies the name and library of the file that is sent. The file being sent can be a physical file or a save file; logical files and device files are not allowed. Overrides to the specified file are ignored.

This is a required parameter.

The possible library values are:

*LIBL   The library list is used to locate the file.

*CURLIB

The current library for the job is used to locate the file. If no library is specified as the current library for the job, QGPL is used.

library-name

Specify the library where the file is located.

User ID (TOUSRID)

Specifies the two-part user ID of one or more users to whom the file is being sent, or the name of one or more distribution lists containing the user IDs of one or more users to whom the file is being sent. A combination of user IDs and distribution lists can be specified on the same command. Each user ID or distribution list is specified as a two-part name, and both parts are required.

This is a required parameter.

You can enter multiple values for this parameter.

Member (MBR)

Specifies the member that is sent from the file. A member name is not allowed if the file is a save file.

*FIRST

The first member (in order of creation) in the file is sent.

member-name

Specify the name of the file member that is sent.

To file type (TOTYPE)

Specifies, when you send a source file, whether the sequence numbers and date fields are to be removed from the transmitted copy of the file. The source file is not changed. This parameter is not valid for non-source files.

The possible values are:

*FROMFILE
> The file type of the source file is used when sending the file. The transmitted file does not change.

*DATA
> The file is sent as a non-source file. The transmitted copy is sent without sequence numbers and date fields.

VM/MVS class (CLASS)

Specifies the VM/MVS SYSOUT class for distributions sent to a VM host system or to an MVS host system.

The possible values are:

A   The class is A.

B-Z, 0-9
> Specify the distribution class. Valid values range from B through Z and 0 through 9.

Send format (FORMAT)

Specifies the record format in which the network file is transmitted.

The possible values are:

*V   The file is sent using variable-length records with trailing blanks removed from each record.

*F   The file is sent as fixed-length records with no trailing blanks removed from the records. Specifying this value affects only network files sent to a System/370. This value is not recommended for sending files to another AS/400 system.

*U   If the file contains null capable fields, specifying this value causes SNDNETF to allow sending of the file and to send the mapping information which indicates which fields are null capable. This value for FORMAT will only produce the desired results when the file is sent to another AS/400 with OS/400 release VRM420 and later installed.

> **Note:** Specifying FORMAT(*F) may increase the amount of storage and time required when transmitting the network file. Specifying FORMAT(*U) also requires that the receiving system must have a database file created with the identical field mappings as the file being sent, in order to receive the file correctly.

Send priority (PTY)

Specifies the queuing priority used for this file when it is being routed through a SNADS network.

The possible values are:

*NORMAL
> The file is sent with a service level priority of data low, which is used for most data traffic. On an AS/400 system, data low distributions are placed on the normal distribution queue specified for the route.

*HIGH
> The file is sent with a service level priority of data high, which is used for high priority data traffic. On an AS/400 system, data high distributions are placed on the data high distribution queue specified for the route.

Examples

Example 1: Sending a Member

```
SNDNETF   TOUSRID((JONES SYSTEM1))  FILE(EMPLOYEE)  MBR(PGMR)
```

This command sends member PGMR of file EMPLOYEE to the user identified to the network with a user ID of (JONES SYSTEM1). The library list is used to locate the file.

Example 2: Sending a Nonsource File

```
SNDNETF   TOUSRID((JONES SYSTEM2))  FILE(EMPLOYEE)  MBR(PGMR)
          TOTYPE(*DATA)
```

This command sends member PGMR of file EMPLOYEE to the user identified to the network with a user ID of (JONES SYSTEM2). The library list is used to locate the file. The file is being sent as a nonsource file removing the sequence numbers and date fields.

Error messages

*ESCAPE Messages

CPF8057
> File &1 in &2 not a physical file or save file.

CPF8058
> File &1 is a spooled file.

CPF8059
> Member name not allowed for save file.

CPF8063
> Cannot assign necessary resource.

CPF8064
> File &1 in &2 member &3 not sent to any users.

CPF8066
> One or more user identifiers on this command is not correct.

CPF8068

Error detected while processing file to be sent.

CPF8072

Object to be sent is greater than maximum size of 2GB.

CPF9005

System resource required to complete this request not available.

CPF9006

User not enrolled in system distribution directory.

CPF9803

Cannot allocate object &2 in library &3.

CPF9807

One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

CPF9810

Library &1 not found.

CPF9812

File &1 in library &2 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

CPF9830

Cannot assign library &1.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9848

Cannot open file &1 in library &2 member &3.

CPF9849

Error while processing file &1 in library &2 member &3.

Top

Send Network Message (SNDNETMSG)

The Send Network Message (SNDNETMSG) command sends a message to another user on the local or a remote system through the SNADS network.

This message is sent as an informational message to the message queue that is defined for the recipient on the receiving system.

Note: The recipient must have a valid message queue specified in his user profile. Messages sent with the SNDNETMSG command are rejected if the recipient does not have a message queue specified in the user profile. The message queue specified in the network attributes is not used. Additional information on specifying a message queue when sending and receiving messages is in the CL information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

This command does not perform the CCSID translation of the messsage. However, it sends the job CCSID to the receiving system.The message is translated upon receipt if needed. The user ID and address of both the recipient and the originator are translated from the current job CCSID to the multinational character set 697/500.

Restriction: The user must be enrolled in the system distribution directory. A description of the system distribution directory is in the SNA Distribution Services book, SC41-5410.

Parameters

Keyword	Description	Choices	Notes
MSG	Message text	*Character value*	Required, Positional 1
TOUSRID	User ID	Values (up to 50 repetitions): *Element list*	Required, Positional 2
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	

Message text (MSG)

Specifies the message text of the immediate message that is sent. An immediate message is a message that is not stored in a message file. The text must be enclosed in apostrophes if it contains blanks or special characters. A maximum length of 256 characters can be specified.

This is a required parameter.

User ID (TOUSRID)

Specifies one or more user IDs, or the name of one or more distribution lists containing user IDs of users to whom the message is to be sent.

A combination of both user IDs and distribution lists can be specified on the same command. Each user ID or distribution list is specified as a two-part name, and both parts are required. Up to 50 user IDs can be specified.

Note: Depending on the type of work station being used, the internal value for a user identifier may differ from the characters shown by the Display Directory Entries (DSPDIRE) command. If the byte-string value specified for the TOUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

This is a required parameter.

You can enter multiple values for this parameter.

Top

Examples

```
SNDNETMSG   MSG('I''m updating the accounts receivable files.')
            TOUSRID((SMITH SYSTEM2))
```

This command sends the message specified in the MSG parameter to the user identified to the network with a user ID of (SMITH SYSTEM2).

Top

Error messages

*ESCAPE Messages

CPF8063
> Cannot assign necessary resource.

CPF8066
> One or more user identifiers on this command is not correct.

CPF8069
> Message not sent to any users.

CPF9005
> System resource required to complete this request not available.

CPF9006
> User not enrolled in system distribution directory.

CPF9830
> Cannot assign library &1.

CPF9845
> Error occurred while opening file &1.

CPF9846
> Error while processing file &1 in library &2.

CPF9847
> Error occurred while closing file &1 in library &2.

Send Network Spooled File (SNDNETSPLF)

The Send Network Spooled File (SNDNETSPLF) command sends a spooled file to another user on the local system or on a remote system on the SNADS network. The file is placed on the output queue that is specified in the user profile of the user to whom the spooled file was sent.

When the file arrives at the destination system, a message is sent to both the recipient and sending user notifying them of the arrival of the spooled file.

This command does not perform any CCSID translation on the contents of the file. However, the user ID and address of both the recipient and the originator are translated from the current job CCSID to the multinational character set 697/500.

Restrictions:

1. The user must be enrolled in the system distribution directory to run this command. The sender must have read, add, and delete authority to the receiving output queue when sending to user on the same system.

2. One of the following must be true:
   - The requester is the creator of the file.
   - The requester has *READ authority to the output queue on which the file resides, and DSPDTA(*YES) was specified on the CRTOUTQ command.
   - The requester has *SPLCTL special authority.
   - The requester has *JOBCTL special authority, and the output queue on which the file resides has OPRCTL(*YES) specified on the CRTOUTQ command.
   - The output queue has DSPDTA(*YES) specified on the CRTOUTQ command.
   - The requester has owner authority to the output queue on which the file resides and the queue had AUTCHK(*OWNER) and DSPDTA(*YES) or DSPDTA(*NO) specified on the CRTOUTQ command.
   - The requester has *READ, *ADD, and *DELETE authority to the output queue on which the file resides and the queue has AUTCHK(*DTAAUT) and DSPDTA(*YES) or DSPDTA(*NO) specified on the CRTOUTQ command.

3. DTAFMT(*RCDDATA) must be used when sending a spooled file to a release prior to Version 1 Release 3 Modification 0 (V1R3).

Parameters

Keyword	Description	Choices	Notes
FILE	Spooled file	*Name*	Required, Positional 1
TOUSRID	User ID	Values (up to 50 repetitions): *Element list*	Required, Positional 2
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	

Keyword	Description	Choices	Notes
JOB	Job name	Single values: * Other values: *Qualified job name*	Optional, Positional 3
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
SPLNBR	Spooled file number	1-999999, ***ONLY**, *LAST, *ANY	Optional, Positional 4
JOBSYSNAME	Job system name	*Name*, ***ONLY**, *CURRENT, *ANY	Optional
CRTDATE	Spooled file created	Single values: ***ONLY**, *LAST Other values: *Element list*	Optional
	Element 1: Creation date	*Date*	
	Element 2: Creation time	*Time*, ***ONLY**, *LAST	
DTAFMT	Data format	***RCDDATA**, *ALLDATA	Optional
CLASS	VM/MVS class	**A**, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Optional
PTY	Send priority	***NORMAL**, *HIGH	Optional

Spooled file (FILE)

Specifies the name of the spooled file that is sent to the specified user. The file name is the name of the device file that was used by the program to produce the spooled output file.

This is a required parameter.

User ID (TOUSRID)

Specifies the two-part user ID of one or more users to whom the file is being sent, or the name of one or more distribution lists containing the user IDs of one or more users to whom the file is being sent. A combination of user IDs and distribution lists can be specified on the same command. Each user ID or distribution list is specified as a two-part name, and both parts are required. The users in the distribution list may be either remote or local.

Note: Depending on the type of work station being used, the internal value for a user identifier may differ from the characters shown by the DSPDIRE command. If the byte-string value specified for the TOUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

This is a required parameter.

You can enter multiple values for this parameter.

Job name (JOB)

Specifies the name of the job that created the spooled output file whose data records are to be sent.

* The job that issued this command is the job that created the spooled file.

qualified-job-name
Specify the qualified name of the job that created the spooled file. If no job qualifier is given, all jobs currently in the system are searched for the simple job name.

Spooled file number (SPLNBR)

Specifies the number of the spooled output file from the job whose data records are to be sent.

*ONLY
Only one spooled file in the job has the specified file name; therefore, the number of the spooled file is not necessary.

*LAST
The spooled file with the highest number and the specified file name is used.

*ANY The spooled file number is not used to determine which spooled file is used. Use this value when the job system name parameter or the spooled file create date and time parameter is to take precedence over the spooled file number when selecting a spooled file.

spooled-file-number
Specify the number of the spooled file having the specified file name whose data records are to be sent.

Job system name (JOBSYSNAME)

Specifies the name of the system where the job that created the spooled file (JOB parameter) ran. This parameter is considered after the job name, user name, job number, spooled file name, and spooled file number parameter requirements have been met.

*ONLY
There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and spooled file create date and time.

*CURRENT
The spooled file created on the current system with the specified job name, user name, job number, spooled file name, spooled file number, and create date and time is used.

*ANY The job system name is not used to determine which spooled file is used. Use this value when the spooled file create date and time parameter is to take precedence over the job system name when selecting a spooled file.

name Specify the name of the system where the job that created the spooled file ran.

Spooled file created (CRTDATE)

Specifies the date and time the spooled file was created. This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, and job system name parameter requirements have been met.

Single values

*ONLY

> There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.

*LAST

> The spooled file with the latest create date and time of the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used.

Element 1: Creation date

date   Specify the date the spooled file was created.

Element 2: Creation time

*ONLY

> There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date.

*LAST

> The spooled file with the latest create time of the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date is used.

time   Specify the time the spooled file was created.

Data format (DTAFMT)

Specifies the format in which to transmit the spooled file. This parameter is applicable only to spooled print files. It is ignored for diskette files.

Spooled files that contain special device requirements cannot be sent using this command if *RCDDATA is specified on the **Data format** prompt (DTAFMT parameter).

Note: The device requirements are listed as part of the attributes for the spooled file. To view the device requirements, use the WRKSPLFA command or the attributes option on the WRKSPLF command. If any of the device requirements are attributes of the file (if any of the device requirements on the display have a 'Y'), the spooled file cannot be sent. Either specify DTAFMT(*ALLDATA) or copy the spooled file to a database file using the Copy Spooled File (CPYSPLF) command and then use the Send Network File (SNDNETF) command to send the file as a data file.

If *RCDDATA is specified on the **Data format** prompt (DTAFMT parameter), the following attributes of the spooled file are kept:

- File name
- Number of copies
- Characters per inch
- Drawer
- Form type
- IGC data indicator

- Lines per inch
- Page length
- Page width
- Page rotation
- Font name
- Print text
- Diskette label
- Diskette creation and expiration dates
- Diskette code type
- Diskette exchange type

The possible values are:

*RCDDATA
> The spooled file is sent in the existing lower function format. Some functions will be removed from data that is sent in this format because the format does not support advanced functions. Use this format for sending spooled files to a System/36, System/38, or a System/370.

*ALLDATA
> The spooled file is sent as it exists, without loss of attributes. All spooled file attributes required to reproduce the file on the receiving system are also sent. Use the *ALLDATA value to send spooled files to Version 1, Release 3, Modification level 0 or later releases of the AS/400 system.

NOTES:
1. If a file will be printed on the receiving system, it must be printed on the same type of printer as it was intended to be printed on the source system.
2. Spool files that require advanced function printing (AFP) resources may print differently on the receiving system. To assure that spool files are printed in the same manner, the following must occur:
   - Before using this command to send a file to a user on the same system, make sure that the libraries containing non-IBM supplied AFP resources are in the library list.
   - Before using this command to send a file to a different user or system, make sure that the libraries containing non-IBM supplied AFP resources are in the initial library list of the user receiving the spooled files.
3. Use this format to send *LINE, *AFPDS, and *AFPDSLINE printer type device files to System 370.

Top

VM/MVS class (CLASS)

Specifies the VM/MVS SYSOUT class for distributions sent to a VM host system or to an MVS host system.

The possible values are:

A   The class is A.

B-Z, 0-9
> Specify the distribution class. Valid values range from B through Z and 0 through 9.

Top

Send priority (PTY)

Specifies the queuing priority used for this spooled file when it is being routed through a SNADS network.

The possible values are:

*NORMAL
> The spooled file is sent with a service level priority of data low, which is used for most data traffic. On an AS/400 system, data low distributions are placed on the normal distribution queue specified for the route.

*HIGH
> The spooled file is sent with a service level priority of data high, which is used for high priority data traffic. On an AS/400 system, data high distributions are placed on the data high distribution queue specified for the route.

<div align="right">Top</div>

Examples

Example 1: Sending a Spooled File

```
SNDNETSPLF   FILE(QPRINT)  TOUSRID((JDE SYS1))
             JOB(142857/PAPER/PRINT)
             SPLNBR(*LAST)  DTAFMT(*ALLDATA)
```

This command sends the last (most recently created) copy of spooled file QPRINT from job 142857/PAPER/PRINT to the user with a user ID of JDE SYS1. All spooled file functions will be sent.

Example 2: Sending Print Attributes

```
SNDNETSPLF   DTAFMT(*RCDDATA)
```

This command sends a limited set of print attributes.

Example 3: Sending All Print Attributes

```
SNDNETSPLF   DTAFMT(*ALLDATA)
```

This command sends all print attributes. *ALLDATA is only valid when it is sent from one iSeries system to another iSeries system.

<div align="right">Top</div>

Error messages

*ESCAPE Messages

CPF2207
> Not authorized to use object &1 in library &3 type *&2.

CPF3205
> File not created.

CPF3207
> Member not added. Errors occurred.

CPF3303
> File &1 not found in job &5/&4/&3.

CPF3309

       No files named &1 are active.

CPF3330

       Necessary resource not available.

CPF3340

       More than one file with specified name found in job &5/&4/&3.

CPF3342

       Job &5/&4/&3 not found.

CPF3343

       Duplicate job names found.

CPF3344

       File &1 number &8 no longer in the system.

CPF3429

       File &1 number &7 cannot be displayed, copied, or sent.

CPF3482

       Copy request failed. Spool file &1 is open.

CPF3486

       CHLVAL parameter value not valid.

CPF3492

       Not authorized to spooled file.

CPF3493

       CTLCHAR parameter not correct for file &1.

CPF3499

       Records in file &1 preceded all assigned channel values.

CPF8055

       Spooled file contains special device requirements. File not sent.

CPF8063

       Cannot assign necessary resource.

CPF8066

       One or more user identifiers on this command is not correct.

CPF8067

       File &1 not sent to any users.

CPF8068

       Error detected while processing file to be sent.

CPF8072

       Object to be sent is greater than maximum size of 2GB.

CPF9005

       System resource required to complete this request not available.

CPF9006

       User not enrolled in system distribution directory.

CPF9820

       Not authorized to use library &1.

CPF9830

       Cannot assign library &1.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9849

Error while processing file &1 in library &2 member &3.

Top

Send Program Message (SNDPGMMSG)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Yes

The Send Program Message (SNDPGMMSG) command sends a message to a named message queue or to a call message queue. A call message queue can be the *EXT external message queue or a message queue associated with a call stack entry. Each time a program or procedure is called a new message queue is associated with its call stack entry. The message queue is identified by the name of its associated program or procedure.

A program can send a message to its own message queue or to a message queue that is associated with a different call stack entry.

This command can send both exception and non-exception messages.

Restrictions:

1. The SNDPGMMSG command allows a message of up to 512 characters to be sent. However, if the message is sent to the *EXT message queue of an interactive job, only 76 characters are shown on the Display Program Messages display. If the message is sent to a user's, work station's, or the system operator's message queue, the Display Message (DSPMSG) command allows all 512 characters to be displayed.

2. This command can only send inquiry messages (specified by MSGTYPE(*INQ)) to one message queue or to two nonprogram message queues if one of the queues is *HSTLOG.

Parameters

Keyword	Description	Choices	Notes
MSG	Message text, or	*Character value*	Optional, Positional 1
MSGID	Message identifier	*Name*	Optional, Positional 2
MSGF	Message file	*Qualified object name*	Optional, Positional 3
	Qualifier 1: Message file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MSGDTA	Message data field values	*Character value*, ***NONE**	Optional, Positional 4
TOPGMQ	Call stack entry message queue	Single values: *EXT Other values: *Element list*	Optional
	Element 1: Relationship	***PRV**, *SAME	
	Element 2: Call stack entry identifier	*Element list*	
	Element 1: Call stack entry	*Character value*, <u>*</u>	
	Element 2: Module	*Name*, ***NONE**	
	Element 3: Bound program	*Name*, ***NONE**	

Keyword	Description	Choices	Notes
TOMSGQ	Send to non-pgm message queue	Single values: ***TOPGMQ**, *SYSOPR Other values (up to 50 repetitions): *Qualified object name*	Optional
	Qualifier 1: Send to non-pgm message queue	*Name*, *HSTLOG	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TOUSR	To user profile	*Name*, *SYSOPR, *ALLACT, *REQUESTER	Optional
MSGTYPE	Message type	***INFO**, *INQ, *RQS, *COMP, *DIAG, *NOTIFY, *ESCAPE, *STATUS	Optional
RPYMSGQ	Message queue to get reply	Single values: ***PGMQ** Other values: *Qualified object name*	Optional
	Qualifier 1: Message queue to get reply	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
KEYVAR	CL var for KEYVAR (4)	*Character value*	Optional
CCSID	Coded character set ID	1-65535, *HEX, ***JOB**	Optional

Message text, or (MSG)

Specifies the message text that is to be sent. A maximum of 3000 characters can be specified or, if you are prompting for this command in an interactive job, a maximum of 512 characters can be specified. The string must be enclosed in apostrophes if special characters (including blanks) are used. If this parameter is specified, a value cannot be specified for the **Message identifier (MSGID)** parameter, and *ESCAPE, *NOTIFY, or *STATUS cannot be specified for the **Message type (MSGTYPE)** parameter. If this parameter is specified, a value cannot be specified for the **Message file (MSGF)** parameter or the **Message data field values (MSGDTA)** parameter, because these types require that a message identifier also be specified.

Coded Character Set Identifier (CCSID) Considerations

The text supplied for the MSG parameter is assumed to be in the CCSID of the job running this command unless a coded character set identifier is supplied in the CCSID parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Message identifier (MSGID)

Specifies the message identifier of a message description whose predefined message is being sent by the program to a message queue. If this parameter is specified, a value cannot be specified for the **Message text, or (MSG)** parameter.

Message file (MSGF)

Specifies the message file that contains the predefined message to be sent. This parameter is required if a value is specified for the **Message identifier (MSGID)** parameter.

Qualifier 1: Message file

name   Specify the name of the message file which contains the predefined message to be sent.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
> The current library for the job is used to locate the message file. If no current library entry exists in the library list, QGPL is used.

name   Specify the library where the message file is located.

Message data field values (MSGDTA)

Specifies the character string, or a CL variable that contains a character string, containing one or more substitution values that are used as message data fields within the predefined message. The substitution values take the place of the substitution variables that were defined in the message text when the message was defined.

*NONE
> There are no program-supplied substitution values used in the specified message.

character-string
> Specify the character string that gives the substitution values in the specified predefined message that is sent by the program, or specify the name of the CL variable that contains the character string.

> **Coded Character Set Identifier (CCSID) Considerations**

> The text supplied for the MSGDTA parameter that corresponds to the *CCHAR type field is assumed to be in the CCSID of the job running this command unless the coded character set identifier is supplied in the CCSID parameter. All other text supplied for the MSGDTA parameter is assumed to be 65535 and is not converted. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter. For more information about the *CCHAR type fields, see the Add Message Description (ADDMSGD) command.

Call stack entry message queue (TOPGMQ)

Specifies the call message queue to which the specified message is to be sent. The message queue can be the *EXT external queue or the call message queue associated with a call stack entry.

Single values

*EXT   The message is sent to the external message queue of the job. The external message queue is used to communicate with the external requester of the job, such as a display station user. *INQ messages that are sent to *EXT wait for 24 hours before the default reply is sent.

Messages sent to this queue can be 512 characters in length, but only 76 characters of text are shown on the Program Messages display.

Element 1: Relationship

Two parameter elements are used to specify the call stack entry message queue from which a message is to be removed. The first element specifies whether the message queue is associated with the program or procedure identified by the second element, or if it is associated with the caller of the program or procedure.

*PRV   The message is sent to the message queue of the call stack entry that is immediately previous to the one identified by the second element of this parameter. However, if the message queue immediately previous to the one identified by the second element is for an Integrated Language Environment (ILE) program entry procedure (PEP), the message is sent to the message queue that precedes the PEP message queue in the stack.

*SAME
   The message is sent to the message queue of the call stack entry identified by the second element of this parameter.

Element 2: Call stack entry identifier

The second element of this parameter has three elements. Element 1 specifies an OPM program or ILE procedure name or a special value. Element 2 specifies an ILE module name which is used as a qualifier for the value specified in element 1. Element 3 can specify either an OPM program name or an ILE program name or a service program name, depending on what is specified in element 1. Element 3 is also used as a qualifier for what is specified in element 1.

Element 1: Call stack entry

*   Specifies the OPM program or ILE procedure running this command.

name Specify the name of the OPM program or ILE procedure used to identify the call stack entry.

        If this element identifies an OPM program, the name specified can be a maximum of 10 characters. If this element identifies an ILE procedure, the name specified can be a maximum of 256 characters.

        Nested procedure names can be specified by separating each procedure name with a colon (:). When specifying nested procedure names, the outermost procedure name is identified first, followed by its contained procedures. The innermost procedure name is identified last in the string.

        Partial names of programs or procedures can be specified by placing three less-than symbols (<<<) at the beginning of the name or by placing three greater-than symbols (>>>) at the end of the name. If both the greater-than symbols and the less-than symbols are used, the program or procedure name specified is limited to 250 characters.

        The system begins its search for the specified program or procedure name with the most recently called program or procedure.

        When searching for a partial program or procedure name:

        • The less-than symbols (<<<) are truncated when specified only at the beginning of a program or procedure name and the remaining character string is right-justified. The remaining characters in the specified string are compared to the current program or procedure on the call stack, starting with the last position of the program or procedure name and comparing backward.

- The greater-than symbols (>>>) are truncated when specified only at the end of a program or procedure name. The remaining characters in the specified string are compared to the current program or procedure on the call stack, starting with the first position of the program or procedure name.
- The less-than symbols (<<<) and the greater-than symbols (>>>) are truncated when both are specified for a program or procedure name. The remaining characters are used to scan and compare the entire length of the specified string with the current program or procedure on the call stack.

Element 2: Module

*NONE
> No ILE module qualifier is provided.

name Specify the ILE module name to be used to identify the message queue.

Element 3: Program

*NONE
> No program qualifier is provided.

name Specify the program name to be used to identify the message queue.

The procedure name alone may not identify the correct procedure. Several different procedures with the same name can run in a job. To further identify a procedure, the name specified can be qualified by a module name, or by both a module name and a bound program name. The following special values can be specified for the first qualifier of the second element of this parameter:

*CTLBDY
> Specifies the call stack entry that is at the most recent control boundary. This entry will be running in the same activation group as the CL program that is running the SNDPGMMSG command. Note that a control boundary will not exist if all programs on the call stack are OPM programs.

*PGMBDY
> Specifies the program boundary of either the program that is using the SNDPGMMSG command or the program whose name is specified for qualifier 3 of this parameter. If no name is specified for qualifier 3, it is assumed that the program is the one using the command.
>
> If it is an ILE program that is being specified, this special value identifies the call stack entry for the program entry procedure (PEP) of that program, if the program was called by a dynamic call. If the program was called by a procedure pointer, this special value identifies the call stack entry for the procedure that was pointed to. If it is an ILE service program that is being specified, this special value identifies the call stack entry for the first procedure that was called in that service program.
>
> If the program being specified is an OPM program, this special value has the same effect as specifying the special value * or a program name for item 1. A difference will occur if the OPM program has called itself recursively. In this case, this special value identifies the first recursion level rather than the current recursion level as would be the case if the special value * or a program name was used.

*PGMNAME
> Specifies that the call stack entry will be identified only by using a program name and optionally

a module name. When this special value is used, qualifier 3 must specify an ILE program or service program name or OPM program name. Qualifier 2 may contain either the special value *NONE or an ILE module name.

This special value is used to send a message to the most recently called procedure that is part of the specified ILE program or service program. When using this special value, it is not necessary to explicitly provide a procedure name. If a module name is also provided, then this special value is used to send a message to the most recently called procedure that is both part of the identified program and the identified module.

This special value may also be used to send a message to an OPM program. In this case, using this special value and providing the OPM program name in item 3 has exactly the same effect as providing that program name here in item 1. Note that if this special value is being used to send to an OPM program then the module name must be specified as *NONE.

Send to non-pgm message queue (TOMSGQ)

Specifies up to 50 nonprogram message queues to which an informational message is sent. For an inquiry message, one message queue may be specified or two message queues may be specified if one of the queues is *HSTLOG. This parameter cannot be used if a value is specified for the **To user profile (TOUSR)** parameter.

Single values

*TOPGMQ
> The message is sent only to the call message queue specified for the **Call stack entry message queue (TOPGMQ)** parameter.

*SYSOPR
> The message is sent to the system operator message (message queue QSYSOPR in library QSYS). Any message sent to message queue QSYSOPR in library QSYS automatically has a copy of the message sent to the QHST (history log) message queue in library QSYS.

Qualifier 1: Send to non-pgm message queue

*HSTLOG
> The message is sent to the system history log (message queue QHST in library QSYS). If *HSTLOG is specified more than once, only one message will be sent to the system history log. If *HSTLOG is specified with message queue QSYSOPR, only one message is sent to the system history log.

name   Specify the name of the message queue to which the message is to be sent. A maximum of fifty message queues can be specified.

Qualifier 2: Library

*LIBL   All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB
> The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

name   Specify the library where the message queue is located.

To user profile (TOUSR)

Specifies that the message is to be sent to the message queue specified in the user profile for the user named on this parameter. This parameter cannot be used if a value is specified for the **Send to non-pgm message queue (TOMSGQ)** parameter or the **Call stack entry message queue (TOPGMQ)** parameter.

*SYSOPR
> The message is sent to the system operator (message queue QSYSOPR in library QSYS). Any message sent to message queue QSYSOPR in library QSYS automatically has a copy of the message sent to the QHST (history log) message queue in library QSYS.

*REQUESTER
> The message is sent to the user profile message queue for interactive jobs or to the system operator's message queue (QSYSOPR in library QSYS) for batch jobs.

*ALLACT
> A copy of the message is sent to the user profile message queue of each user profile with an interactive job currently running. *ALLACT cannot be specified with inquiry messages.

name Specify the user profile name of the user to whom the message is to be sent.

Message type (MSGTYPE)

Specifies which message type is assigned to this message when it is sent by this program.

Notes:

1. Inquiry messages can be sent only to the external queue or to a named message queue specified for the TOUSR or TOMSGQ parameters. When sending an inquiry with the TOMSGQ parameter, a second queue can be specified if the value is *HSTLOG.
2. Completion, diagnostic, escape, notify, and status messages can be sent only to a call message queue.
3. Escape messages cannot be sent to the external message queue.

*INFO
> The message is sent as an informational message.

*INQ The message is sent as an inquiry message.

*COMP
> A completion message is sent to a call message queue. A completion message indicates the status of the work that is successfully performed.

*DIAG
> A diagnostic message is sent to a call message queue. Diagnostic messages provide information about errors detected by this program. The errors are either in the input sent to it, or are those that occurred while it was running the requested function. An escape or notify message should also be sent to inform the receiving program or procedure of the diagnostic messages that are on its message queue.

*NOTIFY
> A notify exception message is sent to a call message queue. A notify message describes a condition for which corrective action must be taken before the sending program can continue. A reply message is sent back to the sending program. After corrective action is taken, the sending program can resume running and can receive the reply message from its message queue.

*ESCAPE
> An escape exception message is sent to a call message queue. An escape message describes an irrecoverable error condition. The sending program does not continue to run.

*RQS A request message is sent to a call message queue. A request message allows request data received from device files to pass from this program to another program or procedure. An immediate message, specified by the MSG parameter, must be used to send the request.

*STATUS

A status exception message is sent to a call message queue. The status message describes the status of work performed by the sending program. The first 28 characters of message data in the MSGDTA parameter are used as the comparison data for message monitors (established by the Monitor Message (MONMSG) command). If the status exception message is not being monitored, control is returned to the sending program. If a status message is sent to the external message queue of an interactive job, the message is shown on line 24, processing continues, and no response is required.

Note: This value cannot be specified if the **Message text, or (MSG)** parameter, is specified.

Top

Message queue to get reply (RPYMSGQ)

Specifies, for inquiry and notify messages only, the call message queue or the non-program message queue to which the reply message is to be sent.

Single values

*PGMQ

The reply to an inquiry or notify message is sent to the message queue associated with the call stack entry of the program or procedure using this command.

Qualifier 1: Message queue to get reply

name Specify the name of the message queue to which the reply is sent.

Qualifier 2: Library

*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*CURLIB

The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

name Specify the library where the message queue is located.

Top

CL var for KEYVAR (4) (KEYVAR)

Specifies the name of the CL character variable, if any, that contains the message reference key that identifies the message sent by the program containing this command. The message reference key is assigned by the system when the message is sent and is placed in the variable specified here.

If a message is being sent to a message queue associated with a call stack entry, KEYVAR refers to that message queue (specified for the **Call stack entry message queue (TOPGMQ)** parameter). If *INQ or *NOTIFY is specified for the **Message type (MSGTYPE)** parameter, KEYVAR refers to the message queue specified for the **Message queue to get reply (RPYMSGQ)** parameter. In all other cases, KEYVAR refers to the message queue specified for the TOPGMQ parameter.

Any type of message can be assigned a key when it is being sent to a program message queue. For messages sent to a nonprogram message queue, message reference keys are available for inquiry (*INQ) messages only. If another message type is sent to a nonprogram queue, no message key is available and blanks are returned for KEYVAR.

The variable must be a character variable having a length of 4 characters. If KEYVAR is not specified and a reply is required, it can be received by the program in FIFO order.

Coded character set ID (CCSID)

Specifies the coded character set identifier (CCSID) that the supplied message or message data is in. If a message identifier is specified, the text supplied by the MSGDTA (message data) parameter that corresponds to the *CCHAR type field is assumed to be in the CCSID specified by the CCSID parameter. The data supplied that does not correspond to the *CCHAR type field is assumed to be 65535 and is not converted. For more information about the *CCHAR type field see the Add Message Description (ADDMSGD) command.

If no message identifier is specified, the text supplied by the MSG (message) parameter is assumed to be in the CCSID supplied by the CCSID parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

*JOB   The message data or immediate text is assumed to be in the CCSID of the job running this command.

*HEX   The message data or immediate text is not converted. CCSID 65535 is used.

coded-character-set-identifier
>   Specify a valid CCSID in which you want your message or message data to be considered in. Valid values range from 1 through 65535. This command validates the CCSID. See the Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for a list of valid CCSID values.

Examples

Example 1: Specifying Substitution Values

```
SNDPGMMSG   MSGID(UIN0023)  MSGF(INV)  MSGDTA('50 100')
            TOPGMQ(*EXT)
```

This command sends the message identified as UIN0023, which is stored in message file INV, to the external message queue of the job (the Display Program Messages presents the message at a display station). The data, which contains two substitution values specified in the MSGDTA parameter, is sent with the message. This data can then be used as substitution values when the message is received, or it can be used as data to be dumped, depending on how the message UIN0023 is defined in the message file. Assuming that the variables &1 and &2 have been defined in the message file as character variables, each 3 characters long, and that the first-level message text of the message UIN0023 is: 'Requested item decreased by &1; current balance &2.' The message text sent is: 'Requested item decreased by 50; current balance 100.'

Example 2: Sending an Inquiry Message

```
SNDPGMMSG   MSG('Mount checks in printer before continuing')
            MSGTYPE(*INQ)  TOMSGQ(*SYSOPR)
```

This command sends an inquiry message to the system operator. The operator looks at the message that was sent by using the DSPMSG command and responds to the message directly on that display. A Receive Message (RCVMSG) command is used in the program to accept the operator's response.

Example 3: Sending an Escape Message

```
SNDPGMMSG   MSGID(USR0001)  MSGF(USRMSGR)  TOPGMQ(*PRV  *)
            MSGTYPE(*ESCAPE)
```

This command is an example of how a message could be sent to the caller of a program or procedure to cause an abnormal end. The message USR0001 could indicate that an invalid code was passed (such as a nonnumeric code when numeric is required). Because the message being sent is an escape message, the program or procedure that is sending the message cannot be resumed. The values *PRV and * did not have to be coded on this command because they are the default values on the TOPGMQ parameter.

Example 4: Sending an Escape Message to an ILE Procedure

```
SNDPGMMSG   MSGID(USR0001)  MSGF(USRMSGR)
            TOPGMQ(*SAME ACCOUNT_FINAL_TOTALS)
            MSGTYPE(*ESCAPE)
```

This command sends a message to an ILE procedure. In this example, the call stack entry identifier is more than 10 characters. Since no qualifier is specified, the actual module name and bound program name associated with the procedure are not used in finding the procedure. The escape exception message is sent to the message queue associated with ACCOUNT_FINAL_TOTALS because *SAME is specified for Element 1.

Example 5: Sending an Escape Message using Qualifiers

```
SNDPGMMSG   MSGID(USR0001)  MSGF(USRMSGR)
            TOPGMQ(*PRV FIRST_QTR_SUMMARY SUMQTRS REPORTS)
            MSGTYPE(*ESCAPE)
```

This command sends an escape exception message to the caller of the procedure FIRST_QTR_SYMMARY. The procedure is qualified by the module name SUMQTRS and the bound program name REPORTS. The escape exception message interrupts the sending program and the sending program is not resumed.

Example 6: Sending a Completion Message using a Partial Procedure Name

```
SNDPGMMSG   MSGID(USR0001)  MSGF(USRMSGR)
            TOPGMQ(*SAME 'MANAGE_SALES>>>')  MSGTYPE(*COMP)
```

This command sends a completion message to the most recent procedure whose name begins with MANAGE_SALES.

Top

Error messages

*ESCAPE Messages

CPF24CB
>   *PGMNAME requires a specified program name.

CPF2409
>   Specified message type not valid with specified program message queue.

CPF2428
>   Message queue parameter is not valid.

CPF2453
>   Reply queue not sender's program message queue.

CPF2469

Error occurred when sending message&1.

CPF247A

Call stack entry not found.

CPF247E

CCSID &1 is not valid.

CPF2499

Message identifier &1 not allowed.

CPF2524

Exception handler not available because of reason code &1.

CPF2550

Exception message sent to a deleted program or procedure.

CPF2702

Device description &1 not found.

CPF7C08

No support network connection.

CPF8C0C

Content of problem record &1 not valid.

CPF8C0E

Library QGPL not found.

CPF8C01

Cannot connect to IBM service system. One session allowed.

CPF8C07

A parameter is not valid.

CPF8C08

Cannot specify *SELECT for the control point name.

CPF8C09

&1 not defined as a service provider.

CPF8C16

Error occurred while processing request.

CPF8C17

Sign-on failed.

CPF8C18

No support network connection.

CPF8C19

Remote support application failed.

CPF8C2A

Cannot connect to IBM service system.

CPF8C24

Error occurred while processing request.

CPF8C27

Alternate load device not found.

CPF8C32

PTF order cannot be processed.

CPF9830

      Cannot assign library &1.

CPF9845

      Error occurred while opening file &1.

CPF9846

      Error while processing file &1 in library &2.

CPF9847

      Error occurred while closing file &1 in library &2.

Top

Send PTF Order (SNDPTFORD)

Where allowed to run: All environments (*ALL)		
Threadsafe: No		

The Send Program Temporary Fix Order (SNDPTFORD) command allows you to prepare an order for:

- Individual PTFs
- Cumulative PTF package
- PTF Group
- Summary information for available PTFs
- Preventive Service Planning (PSP) information

Restriction:

1. This command is shipped with public *EXCLUDE authority, and the QSRV and QSRVBAS user profiles have private authority to use the command.

2. The following restrictions apply for the IMGDIR parameter:

- You must have *X authority to each directory in the path.
- You must have *WX authority to the directory that contains optical image.

Parameters

Keyword	Description	Choices	Notes
PTFID	PTF description	Values (up to 20 repetitions): *Element list*	Required, Positional 1
	Element 1: PTF identifier	*Character value*, *CUMPKG, *HIPERGRP, *DB2GRP, *BRSGRP, *JVAGRP, *HTTPGRP, *PFRGRP	
	Element 2: Product	*Character value*, ***ONLYPRD**	
	Element 3: Release	*Character value*, ***ONLYRLS**	
PTFPART	PTF parts	***ALL**, *CVRLTR	Optional
RMTCPNAME	Remote control point	*Communications name*, ***IBMSRV**, *SELECT	Optional
RMTNETID	Remote network identifier	*Communications name*, ***NETATR**	Optional
DELIVERY	Delivery method	***LINKONLY**, *ANY	Optional
DLVRYFMT	Delivery format	***SAVF**, *IMAGE	Optional
ORDER	Order	***REQUIRED**, *PTFID	Optional
REORDER	Reorder	***NO**, *YES	Optional
CHKPTF	Check PTF	***NO**, *YES	Optional
IMGDIR	Image directory	*Path name*, ***DFT**	Optional
IMGPFX	Image prefix	*Character value*, ***DFT**	Optional

PTF identifier (PTFID)

Specifies the list of PTFs being ordered.

Element 1: PTF identifier

*CUMPKG
> Order the latest level of the cumulative PTF package (SF99vrm) for the operating system release that is installed on the system. HIPER and DB2 Universal Database (UDB) group PTFs are automatically requested with this order. It cannot be ordered with any other PTF identifier or special value.

*HIPERGRP
> Order the latest level of HIPER PTF group for the operating system release that is installed on the system.

*DB2GRP
> Order the latest level of DB2 UDB PTF group for the operating system release that is installed on the system.

*BRSGRP
> Order the latest level of Backup Recovery Solutions PTF group for the operating system release that is installed on the system.

*HTTPGRP
> Order the latest level of IBM HTTP Server PTF group for the operating system release that is installed on the system.

*JVAGRP
> Order the latest level of Java PTF group for the operating system release that is installed on the system.

*PFRGRP
> Order the latest level of Performance Tools PTF group for the operating system release that is installed on the system.

character-value
> Specify the PTF identifier. Some PTFs must be ordered individually or within a list of PTFs with the same prefix and not as part of a general list.
>
> A cumulative PTF package is specified using the format SF99vrm and the Preventive Service Planning (PSP) format is SF98vrm, where vrm is version-release-modification.

Element 2: Product

*ONLYPRD
> The PTF identifier is associated with only one product.
>
> **Note:** If this value and ORDER(*REQUIRED) are specified, requisites are sent for only that product that is installed or supported on your system.

character-value
> Specify the 7-character product identifier of the product that the PTF is associated with. If the PTF identifier is associated with more than one product, the PTF order is limited to the product specified.

Element 3: Release

*ONLYRLS
> The PTF identifier is associated with only one release.
>
> **Note:** If this value and ORDER(*REQUIRED) are specified, requisites are sent for only that release level that is installed or supported on your system.

character-value
> Specify the 6-character 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.

PTF parts (PTFPART)

Specifies whether PTFs or cover letters are being ordered.

*ALL   PTFs and cover letters are being ordered.

*CVRLTR
> Cover letters only are being ordered.

Remote control point (RMTCPNAME)

Specifies the remote control point of the service provider to whom the service request is sent.

*IBMSRV
> The service request is sent to IBM service support.

*SELECT
> A list of service providers is shown so the user can select the destination for the service request.

communications-name
> Specify the name of the remote control point.

Remote network identifier (RMTNETID)

Specifies the remote name identifier of the service provider's network.

*NETATR
> The service provider is in the local network.

communications-name
> Specify the network name of the service provider to whom the request is being sent.

Delivery method (DELIVERY)

Specifies how the PTFs are delivered.

*LINKONLY
> PTFs are delivered by the electronic customer support service link only. If the PTFs are too large to use service link, the command will fail.

*ANY   PTFs are delivered by any available method. The service link is used for most PTFs. PTFs that are too large for the service link are sent on the selected medium.

Delivery format (DLVRYFMT)

Specifies the format of the delivered PTFs.

*SAVF PTFs are delivered through the electronic customer support service link as save files.

*IMAGE

> PTFs are delivered through the electronic customer support service link as optical image files. Optical image files will contain PTFs and cover letter. The optical image file will be stored in the directory spacified by the IMGDIR parameter.

Order (ORDER)

Specifies the level of fixes that are being requested.

*REQUIRED

> The PTF ordered and its requisites are being requested.

*PTFID

> The specific PTF ordered is the one being requested. No requisites are sent.

Reorder (REORDER)

Specifies whether a PTF that is currently loaded, applied, or on order should be ordered again.

*NO PTFs that are already loaded, applied, or on order are not reordered.

*YES PTFs that are already loaded, applied, or on order are reordered.

Note: A PTF is not reordered if the *SAVF delivery format is specified and a save file is available on the system.

Check PTF (CHKPTF)

Specifies whether checking is performed on the service requester system to determine if PTFs are ordered based on whether or not the PTF product is installed or supported.

*NO The PTFs specified on the PTF identifier (PTFID) parameter are ordered even when the PTF product is not installed or supported on the service requester.

*YES The PTFs specified on the PTF identifier (PTFID) parameter are ordered only if the PTF product is installed or supported on the service requester.

Image directory (IMGDIR)

Specifies the directory where the optical image files are stored.

*DFT The optical image files are stored in /QIBM/UserData/OS/Service/ECS directory.

path-name

> Specify an existing directory. For more information on specifying path names, refer to "Object

naming rules" in "CL concepts and reference" in the CL reference information in the iSeries Information Center at http://www.iseries.ibm.com/infocenter.

Image prefix (IMGPFX)

Specifies a prefix for the optical image file names.

*DFT   The files will be named by the Service Provider.

character-value
> Specify the prefix that will be used when naming the optical images being stored. If multiple images are received under one order, the files will be uniquely identified by a numerical suffix on the image name. For more information on specifying file names, refer to "Object naming rules" in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Examples

Example 1: Sending Order for PTFs by PTF Number only

```
SNDPTFORD   PTFID((SI12345) (SI12346))
```

This command sends a request for PTF numbers SI12345 and SI12346.

Example 2: Sending Order for PTF by Product and Release

```
SNDPTFORD   PTFID((SI12345 5722SS1 V5R3M0))
            DELIVERY(*ANY)  ORDER(*REQUIRED)
```

This command sends a request for PTF SI12345 for Version 5 Release 1 Modification 0 of the operating system product (5722SS1). The PTF can be delivered by any available method. Any requisites are sent with the PTFs.

Example 3: Sending Order for Latest Cumulative PTF

```
SNDPTFORD   PTFID((*CUMPKG))
```

This command requests that the latest PTF cumulative package be sent for the operating system release level that is installed on your system.

Example 4: Sending Order for a PTF Group

```
SNDPTFORD   PTFID((SF99893))
```

This command sends a request for PTF group number SF99893.

Example 5: Sending Order for a DB2 UDB Group with *IMAGE Delivery format

```
SNDPTFORD   PTFID((*DB2GRP))  DELIVERY(*LINKONLY)
            DLVRYFMT(*IMAGE)
```

This command sends a request for the latest level of the DB2 UDB PTF group for the operating system release that is installed on your system. The PTF group will be stored in optical image files inside /QIBM/UserData/OS/Service/ECS directory.

Error messages

*ESCAPE Messages

CPF2702

Device description &1 not found.

CPF7C08

No support network connection.

CPF8C0C

Content of problem record &1 not valid.

CPF8C0E

Library QGPL not found.

CPF8C01

Cannot connect to IBM service system. One session allowed.

CPF8C07

A parameter is not valid.

CPF8C08

Cannot specify *SELECT for the control point name.

CPF8C09

&1 not defined as a service provider.

CPF8C16

Error occurred while processing request.

CPF8C17

Sign-on failed.

CPF8C18

No support network connection.

CPF8C19

Remote support application failed.

CPF8C2A

Cannot connect to IBM service system.

CPF8C24

Error occurred while processing request.

CPF8C27

Alternate load device not found.

CPF8C32

PTF order cannot be processed.

CPF8C99

PTF &2-&1 &3 not ordered.

CPF9846

Error while processing file &1 in library &2.

Top

Send/Receive File (SNDRCVF)

Where allowed to run:
- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: No

The Send/Receive File (SNDRCVF) command is used by a CL procedure to send data to and receive data from a device that is being used interactively by a user. The data is passed between the program in which the SNDRCVF command is used and the display device identified in the command. The data is passed using the display device file that was declared in the program. (A Declare File (DCLF) command included in the source used to compile the program was used to declare the file.) The data for each send/receive operation is passed as one record in a format identified by the RCDFMT parameter of this command (the format is defined in the data description specifications (DDS)). One CL variable is used for each field of the record format to pass the data. The CL variables used (including DDS indicators) are declared implicitly.

Of the record formats specified in the DCLF command, only one can be specified in each SNDRCVF command. If the device file has not been opened, it is opened by this command. The file and record format specified in this command can be overridden by an Override with Display File (OVRDSPF) command if that command is entered before the file is opened. However, care should be taken that the fields in the overriding record format correspond to the CL variables declared in the program.

Restrictions: This command is valid only within a CL procedure and only for display files. It cannot be used with database files.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Display device	*Name*, ***FILE**	Optional, Positional 1
RCDFMT	Record format	*Name*, ***FILE**	Optional, Positional 2
OPNID	Open file identifier	*Simple name*, ***NONE**	Optional
WAIT	Wait	***YES**, *NO	Optional

Top

Display device (DEV)

Specifies the name of the display device that the CL procedure's data is to be sent to and the user's data is to be received from. A CL variable can be specified for this parameter so that the device name can be changed without changing the command.

*FILE   The data is to be sent to and received from the device associated with the device file (the device file that was declared in the FILE parameter of the DCLF command). If more than one device name is specified in the device file, *FILE cannot be specified.

name   Specify the name of the device or the name of the CL variable that contains the name of the

835

device that the CL procedure is to send data to and receive data from. If a CL variable name is used in this parameter, only one SNDRCVF command is needed in the program to receive data from several devices.

<div align="right">Top</div>

Record format (RCDFMT)

Specifies the name of the record format that is to be used to pass the data between the CL procedure and the user. The format contains all the fields in the record. This parameter must be coded with a record format name if there is more than one record format in the device file; *FILE cannot be coded if there is more than one. SNDRCVF ignores the INVITE DDS keyword.

*FILE There is only one record format in the device file; that format is to be used to send the data to and receive the data from the user.

name Specifies the name of the record format in which the data is to be sent to and received from the user. A CL variable name cannot be used to specify the record format name.

<div align="right">Top</div>

Open file identifier (OPNID)

Specifies the open file identifier that was declared on a preceding Declare File (DCLF) command in the same CL procedure. A CL variable cannot be specified for this parameter value.

*NONE
No open file identifier is provided. This command will use the file associated with the DCLF command that had *NONE specified for the OPNID parameter. Only one file can be declared in a CL procedure with *NONE as the open file identifier.

simple-name
Specify a name that matches the OPNID parameter value on a preceding DCLF command in the same CL procedure.

<div align="right">Top</div>

Wait (WAIT)

Specifies whether the CL procedure either waits to receive the data from the user's device or continues to process the commands that follow this SNDRCVF command. If WAIT(*NO) is specified, the program must issue a WAIT command later in the program to complete the input operation.

Note: A CL variable cannot be coded on this parameter.

*YES The program waits until the input operation from the device is completed; the next command is not processed until then.

*NO The program does not wait for the input data; it continues to process commands until a WAIT command is reached later in the program.

<div align="right">Top</div>

Examples

Example 1: Using Display File with One Record Format

```
DCLF    FILE(MENU1)
 :
SNDRCVF
```

This command sends and receives user data by way of the device file MENU1. Only one record format exists in the file. The device used is specified in the file.

Example 2: Using Display File with Multiple Record Formats

```
DCLF    FILE(SCR)   RCDFMT(REC8)
 :
SNDRCVF    RCDFMT(REC8)
```

The CL procedure sends data to a user and receives data for the user who is using the device named in the device file (*FILE is assumed because DEV is not specified). The data is passed in the format specified by REC8 record format in the device file named SCR. The CL procedure waits for the user data before continuing.

Example 3: Using a CL Variable for Device Name

```
DCLF    FILE(DF1)   RCDFMT(REC8)
 :
SNDRCVF    DEV(&DN)   RCDFMT(REC8)   WAIT(*NO)
 :
WAIT    DEV(&DN)
```

This command sends and receives user data by way of the device file named DF1. Using the record format REC8, the CL procedure passes data between itself and the user who is at the device named in the variable &DN, but it does not wait for a response to come back. If the procedure sends and receives data from several devices, the same SNDRCVF command can be used. Only the device specified by &DN for the DEV parameter must be changed. A WAIT command for each device must be issued later in the procedure to ensure that all the devices respond.

Example 4: Using Open File Identifier

```
DCLF    FILE(SCREEN1)   RCDFMT(REC1 REC2)   OPNID(OUTDSP1)
DCLF    FILE(SCREEN2)   RCDFMT(REC3 REC4)   OPNID(OUTDSP2)
 :
SNDRCVF    DEV(*FILE)   RCDFMT(REC2)   OPNID(OUTDSP1)   WAIT(*YES)
```

The device file named SCREEN1 is used to send data to the display device named in the same device file and wait for input. The data is presented to the user in the format specified by record format REC2. The SNDRCVF command is associated with device file SCREEN1 because the open file identifier specified on the SNDRCVF command matches the open file identifier specified on the DCLF command for display file SCREEN1.

Top

Error messages

*ESCAPE Messages

CPF0859
> File override caused I/O buffer size to be exceeded.

CPF0861
> File &1 in library &2 is not a display file.

CPF0863
> Value of binary data too large for decimal CL variable.

CPF0864

        End of file detected for file &1 in &2.

CPF0883

        *FILE not valid in DEV parameter for file &1.

CPF0886

        Record contains a data field that is not valid.

CPF0887

        Data available from previous input request.

CPF4101

        File &2 in library &3 not found or inline data file missing.

CPF5068

        Program device &4 not found in file &2 in library &3.

CPF5070

        File &2 in library &3 has no program devices acquired.

Top

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY8809
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) IBM 2006. Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. 1998, 2006. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing
AFP
AS/400
CICS
COBOL/400
C/400
DataPropagator
DB2

IBM
Infoprint
InfoWindow
iSeries
LPDA
OfficeVision
i5/OS
Print Services Facility
RPG/400
SystemView
System/36
TCS
WebSphere

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

IBM ®

Printed in USA