iSeries

CL Commands Volume 16

IBM

eserver

iSeries

CL Commands Volume 16

Contents

Command Descriptions

RCVDST (Receive Distribution) Command Description

RCVDST Command syntax diagram

Purpose

The Receive Distribution command allows a user to receive incoming distributions, such as documents. The documents can be placed in folders, in document objects, or in an output file for processing.

Restrictions:

1. Users cannot receive distributions on behalf of another user unless they have been granted permission to work on behalf of that user with the Grant User Permission (GRTUSRPMN) command.
2. The user ID and address must be entered in the system directory.
3. Private distributions cannot be received if the requester is working on behalf of another user.

Required Parameters

DSTID Specifies the unique identifier of the distribution. The identifier is assigned to the distribution by the system that creates it. Only incoming distributions can be received. If the identifier represents an outgoing distribution, an error message is returned to the user.

The distribution identifier consists of the sender's address (padded on the right with blanks up to 8 characters), the sender's user ID (padded on the right with blanks up to 8 characters), and a 4-digit zoned sequence number with leading zeros, for example,

```
'NEWYORK SMITH   0204'
or
MARYLANDMIKEJONE0099
```

Apostrophes are needed if there are blanks or special characters in the distribution identifier. The distribution identifier is specified this way because blank characters are valid in a user ID or address.

Note: If DSTID is the only parameter for which a value is specified, the distribution specified is deleted from the incoming mail log and confirmation of delivery is sent back to the sender, even if ACKRCV(*NO) is specified.

Optional Parameters

DOC Specifies the document object name under which a distribution is filed when it is received. The document object name that is specified must not exist on the system.

***NONE:** The distribution being received is not filed under a document object name.

document-name: Specify the name of the document object name under which the distribution is filed.

FLR Specifies the name of the folder that contains the document.

Note: The folder must exist and the current user of this command must have authority to create new documents in the folder.

***NONE:** The distribution being received is not placed in a document object. Specify this value if the distributed document is received directly into a database file for processing.

folder-name: Specify the name of the folder to contain the document object name. A folder name can consist of a series of folder names if the document receiving the distribution is located in a folder contained in another folder. Up to 63 characters can be specified.

OUTFILE

Specifies the name of the database file to which the output is directed. If the output file does not exist, this command creates a database file in the specified library. If the file is created by this function, the text will read “Outfile for RCVDST.” The authority for users with no specific authority to use the outfile is *EXCLUDE. More information on defining the format of database files (output files) is in the Office Services Concepts and Programmer’s Guide book.

***NONE:** The output is not directed to a database file.

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

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

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

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

database-file-name: Specify the name of the database file that receives the output.

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

OUTDTATYP

Specifies which distribution data is written to the database file.

***DFT:** The following record codes are written to the output file:

Record Code	Description
010	Distribution Description
020	Message Text
105	Document Description
800	Document Data

***ALL:** All record formats are written to the output file.

Record Code	Description
010	Distribution Description
020	Message Text
105	Document Description
110	Creation Date
115	Expiration date
120	Document date
125	File date
130	Change date
135	Action due date
140	Completion date
145	Author
150	Copy list
155	Document class
160	File cabinet reference
165	Subject
170	Keyword
175	Reference
180	Status
185	Project
190	Authorizing Userid and Address
195	Distribution expiration date and time
200	Reply requested date and time
500	Interchange document profile data
800	Document Data

***DSTINFO:** The distribution description record is written to the output file.

***MSG:** The message text record is written to the output file.

***DOCD:** The document description record is written to the output file.

***DOCDATE:** The document date record is written to the output file.

***FILDATE:** The file date record is written to the output file.

***CRTDATE:** The create date record is written to the output file.

***REF:** The reference record is written to the output file.

***CHGDATE:** The change date record is written to the output file.

***EXPDATE:** The expiration date record is written to the output file.

***ACTDATE:** The action due date record is written to the output file.

***CMPDATE:** The completion date record is written to the output file.

***SUBJECT:** The subject records are written to the output file.

***FILCAB:** The filing cabinet reference record is written to the output file.

***STATUS:** The status record is written to the output file.

***AUTHOR:** The author records are written to the output file.

***DOCCLS:** The document class record is written to the output file.

***KWD:** The keyword records are written to the output file.

***CPYLST:** The copy list records are written to the output file.

***AUTUSR:** The authorizing Userid and Address is written to the output file.

***DSTEXPDATE:** The distribution expiration date and time is written to the output file.

***RPYDATE:** The reply requested date and time is written to the output file.

***IDP:** The complete document record (base sub-profile, application subprofile, and any private sub-profiles) is written to the output file.

***DOC:** The document data records are written to the output file.

***PROJECT:** The project record is written to the output file.

USRID

Specifies the user ID and address of the user for whom the distribution is received. The current user of this command must have the authority to work on behalf of the specified user ID and address.

***CURRENT:** The user profile that is currently running is used.

Element 1: User ID

user-ID: Specify the user ID of the user for whom the distribution is received.

Element 2: User Address

user-address: Specify the user address of the user for whom the distribution is received.

ACKRCV

Specifies whether the acknowledgment for confirmation of delivery is sent back to the sender of the distribution.

***YES:** The confirmation of delivery is sent back to the sender.

***NO:** The confirmation of delivery is not sent back to the sender.

DSTIDEXN

Specifies the extension of the distribution identifier specified on the DSTID parameter. This extension uniquely identifies duplicate distributions. This extension is a 2-digit extension that ranges from 01 through 99. For example, if the distribution ID is *NEWYORK SMITH 0204* and two copies of this distribution were sent to a user, then the user has 2 distributions with the same distribution ID. To distinguish the two distributions, an extension is added to each distribution ID and one extension is *NEWYORK SMITH 020401* and the other one is *NEWYORK SMITH 020402*. If there are no duplicates, the extension defaults to 01. These extensions map one to one with the distribution ID specified on the DSTID parameter.

***NONE:** There is no duplicate distribution. This is equivalent to an extension of 01.

distribution-ID-extension: Specify the extension associated with the distribution. This is used to uniquely identify duplicate distribution IDs.

KEEP Specifies whether this distribution is either deleted from, or kept in, the mail log.

***NO:** When all the information requested has been written to the output file or document object name, the distribution is removed from the user's incoming mail.

***YES:** When all the information requested has been written to the output file or document object name, the distribution is not removed from the user's incoming mail. The incoming distribution is available for another RCVDST request or for processing by another DIA interface, such as OfficeVision.

CMDCHRID

Specifies the character identifier (graphic character set and code page) for data being specified as parameter values on this command. This character identifier (CHRID) is related to the display device used to specify the command. More information about CHRID processing is in the Application Display Programming book.

Note: This value translates the DSTID and USRID parameters to the character set and code page of '930 500'. The SNA Distribution Services book contains the character set and code page table for '930 500'.

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

***DEVD:** The system determines the graphic character set and code page values for the command parameter from the display device description where the command is entered. This option is valid only when specified from an interactive job. If this value is specified in an interactive CL program or a batch job, an error message is sent.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values used to create the command parameter.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Examples of RCVDST

Example 1: Receiving Current User Distribution

```
RCVDST   DISTID('SYSTEM1 USERA 0001')
  OUTFILE(MYLIB/MYFILE)
  OUTMBR(MYMBR *ADD)  OUTDTATYP(*ALL)  CMDCHRID(*DEVD)
```

This command receives the current user distribution into output file MYFILE located in library MYLIB. The distribution is added to member MYMBR. All output file information is added to the output file MYFILE.

Example 2: Receiving Distribution Sent to a User

```
RCVDST   DSTID('BAKER RCH38P 0019')  DSTINDEXN(01)
  OUTFILE(JOWLIB/DOCUMENTS)  USRID(*CURRENT)
```

This command receives a distribution that was sent to a user. It is copied into the first member in a database file called DOCUMENTS in a library called JOWLIB.

Error messages for RCVDST

*ESCAPE Messages

CPF8A87
Document name &2 not correct.

CPF8A97
Folder name &1 not correct.

CPF900B
User ID and address &1 &2 not in System Distribution Directory.

CPF900C
Sign on and verify of user failed.

CPF905C
Error occurred trying to find a translation table.

CPF9096
Cannot use CMDCHRID(*DEVD), DOCCHRID(*DEVD) in batch job.

CPF9098
Distribution not received.

CPF9845
Error occurred while opening file &1.

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

CPF9847
Error occurred while closing file &1 in library &2.

CPF9860
Error occurred during output file processing.

RCVF (Receive File) Command Description

RCVF Command syntax diagram

Purpose

The Receive File (RCVF) command is used within a CL program to receive data from a display device or database file. The command reads a record from the file and puts the data from the record into one or more CL variables. These CL variables were automatically declared in the program when the CL source program was compiled and a Declare File (DCLF) command was processed as part of the source. There is one CL variable for each field in the record format used to receive the data. The data that is entered by a user at the display or is contained in the input record is copied into CL variables in the program by the RCVF command, where it is processed by the program.

Only one record format, of those specified in the DCLF command, can be specified in each RCVF command. If the file has not been opened by a previous RCVF, SNDRCVF, or SNDF command, it is opened by this command. If the file has been previously closed due to an end-of-file condition on a previous RCVF command, an error occurs. The file specified in this command can be overridden if the override command is entered before the file is opened. If the file specified in the DCLF command was a display file when the program was compiled, the file may only be overridden to another display file. If the file was a database file, the file may only be overridden to another database file that has a single record format. However, care should be taken that the fields in the overriding record format correspond to the CL variables declared in the program.

Optional Parameters

DEV Specifies the name of the display device from which data is being received. If a CL variable name is used in this parameter, only one RCVF command is needed in the program to receive data from several devices. (The variable specifying the device name can be changed while repeatedly running the same command.) This parameter may be specified only if the file is a display device file.

***FILE:** The user's data is 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.

device-name: Specify the name of the device or the name of the CL variable that contains the name of the device from which the user's data is being received.

RCDFMT
Specifies the name of the record format that is used to receive data from the file. The format contains all the fields in the record. This parameter must be coded with a record format name if there is more than one record format in the device file. If the file is a database file, the specified record format is used to map the data from the record into the CL variables. The actual record format name in the file at run time may be different. RCVF ignores the INVITE DDS keyword.

***FILE:** There is only one record format in the device file; that is the format in which the data is being received. If more than one record format is specified in the device file, *FILE cannot be specified.

record-format-name: Specify the name of the record format in which the data records from the display device are being received. A CL variable cannot be used to specify the record format name.

WAIT Specifies whether the CL program waits for the data being received from the user's device or continues processing the commands that follow this RCVF command. If WAIT(*NO) is specified, the program must issue a WAIT command later in the program to complete the input operation. This parameter may be specified only if the file is a display device file.

***YES:** The program waits until the input operation from the device is completed; the next command is not processed until then.

***NO:** The program does not wait for the input data; commands continue running until a WAIT command is reached later in the program.

Examples for RCVF

Example 1

```
DCLF   FILE(MENU1)
  *
  *
  *
RCVF
```

The CL program receives data from the user through the file named MENU1. The program waits for the user data before it continues processing.

Example 2

```
DCLF   FILE(SCREENX)  RCDFMT(R1 R2)
  *
RCVF   DEV(DISPLAY2)  RCDFMT(R1)
```

The CL program receives data from the user at the display station named DISPLAY2. The data is received in the record format named R1 in the device file named SCREENX. The program waits for the user data before it continues processing.

Example 3

```
DCLF   FILE(INPUT)
  *
RCVF
MONMSG   CPF0864  EXEC(GOTO EOF)
```

The CL program receives a record sequentially from the database file named INPUT. The program monitors for the end-of-file exception CPF0864 and goes to label EOF when the message is received.

Example 4

```
DCLF   FILE(MSCREEN)  RCDFMT(MIN1 MIN2 MIN3)
  *
  *
  *
RCVF   DEV(&DNAME)  RCDFMT(MIN2)  WAIT(*NO)
WAIT   DEV(&DNAME)
```

The CL program receives user data from several devices one at a time by way of the device file named MSCREEN. The program receives data from the device named in the variable &DNAME using the record format MIN2, but it does not wait for the data to come in. The same RCVF command is used to receive data from several devices; because the CL variable &DNAME is used, only the device name in the DEV parameter must be changed each time the command is run. A WAIT command for each device must be issued later in the program because the WAIT command actually receives the data. Both the RCVF and the WAIT commands may be processed for each device (one at a time) to send data to the program. If a user response is delayed, the commands can be processed as many times as necessary until the user responds with the data or a End Receive (ENDRCV) command cancels the request.

Error messages for RCVF

*ESCAPE Messages

CPF0859
File override caused I/O buffer size to be exceeded.

CPF0860
File &1 in &2 not a data base file.

CPF0861
File &1 in library &2 is not a display file.

CPF0863
Value of binary data too large for decimal CL variable.

CPF0864
End of file detected for file &1 in &2.

CPF0865
File &1 has more than one record format.

CPF0883
*FILE not valid in DEV parameter for file &1.

CPF0886
Record contains a data field that is not valid.

CPF4101
File &2 in library &3 not found or inline data file missing.

CPF502A
Variable length record error on member &4.

CPF502B
Error occurred in trigger program.

CPF502D
Referential constraint violation on member &4.

CPF502E
Referential constraints could not be validated for member &4.

CPF502F
Check constraint violation on member &4.

CPF5029
Data mapping error on member &4.

CPF503A
Referential constraint violation on member &4.

CPF503B
Record could not be inserted or updated in member &4.

CPF5068
Program device &4 not found in file &2 in library &3.

CPF5070
File &2 in library &3 has no program devices acquired.

RCVJRNE (Receive Journal Entry) Command Description

RCVJRNE Command syntax diagram

Purpose

The Receive Journal Entry (RCVJRNE) command allows a specified user exit program to continuously receive journal entries. This program can be set up, for example, to write the entries either (1) to an ICF file, supplying updates to a file on a backup system, or (2) on a tape, imitating a journal-to-tape function. The information in the journal entries received can be used to update the database files being journaled to minimize the loss of data in the event of a disk failure, and to update database files on a backup system in case of a system failure on the primary system.

The value specified on the ENTFMT parameter determines the format of the journal entries passed to the exit program.

Restrictions:

1. If the sequence number is reset in the range of receivers specified, the first occurrence of the FROMENT or TOENT parameter is used if either is specified.
2. The FILE, JRNCDE, ENTTYP, JOB, PGM, USRPRF, CMTCYCID, and DEPENT parameters can be used to specify a subset of all available entries within a range of journal entries.
   - If no values are specified using these parameters, all available journal entries are received.
   - If more than one of these parameters are specified, then a journal entry must satisfy all of the values specified on these parameters, except when *IGNFILSLT 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.
3. The JOB, PGM, and USRPRF parameters cannot be used to specify selection criteria if one or more journal receivers in the specified receiver range was attached to a journal that had » a RCVSIZOPT or FIXLENDTA option specified that omitted the collection of that data. «
4. If more than the maximum number of objects is identified (32,767 objects), an error occurs and no entries are received. This restriction is ignored if *ALLFILE is specified.

Required Parameters

JRN Specifies the qualified name of the journal where the journal entries are to be received.

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

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

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

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

journal-name: Specify the name of the journal where the journal entries are to be received.

EXITPGM

Specifies the qualified name of a user-written exit program that is given control to receive each journal entry passed from the command. Additional information on the interface between this command and the exit program is supplied after the listing of possible values for this parameter, and is described in more detail in the Journal management article in the Information Center.

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

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

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

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

program-name: Specify the name of the exit program that controls the reception of each journal entry passed from the command.

Additional Information on the Exit Program Interface

When the program is called, two parameters are passed to it at a time. A single journal entry or a block of journal entries is passed in the first parameter.

- If a single journal entry is passed, and if the length of the parameter defined by the program is smaller than the length of the journal entry, the journal entry passed to the program is truncated. If the length of the parameter defined by the program is greater than the length of the journal entry, the parameter positions beyond the length of the journal entry contain nonessential information. The user's program should not specifically refer to data in the positions beyond the length of the journal entry.
- » If BLKLEN(*NONE) was specified, then the exit program may indicate to the system that multiple entries should be returned in subsequent calls to the exit program by specifying 8 in the first byte of the second parameter. This is called "Block Mode" and the exit program must specify the size of the block in bytes as a zoned value in the first 5 bytes of the first parameter. If an error is made in this specification, only one journal entry is passed in the block. If BLKLEN(*NONE) was not specified, then specifying 8 in the first byte of the second parameter will have no effect and the first 5 bytes of the first parameter will be ignored. «

- » If pointers are being returned, « this first parameter must be aligned on a 16-byte boundary since journal entry specific data could include actual pointers.

» For *TYPE1, *TYPE2, *TYPE3 and *TYPE4 formats « there is a zoned journal entry length field that is filled with zeros at the end of the single journal entry or block of journal entries passed. This field indicates that the last journal entry has been passed. The format of the information in each journal entry is shown in the ENTFMT parameter description. The format of the first parameter is detailed in the Journal management article in the Information Center.

Note: The maximum length of the parameter specification in the exit program is language dependent (for example, for CL, the maximum length is 9999). For more information about limitations refer to the corresponding programming language book.

A character variable of LEN(3) is passed in the second exit program parameter. This parameter will be passed from the system to the exit program and can be passed from the exit program to the system. Its values are presented in the following lists:

- Information in the first byte of the second parameter:

  Char(1)
  Passed to the Exit Program from the System

  0 No journal entry is passed on this call of the exit program.

  1 A single journal entry is passed to the exit program.

  2 A block of one or more journal entries is passed to the exit program.

  3 No journal entry is passed on this call to the exit program, and no more entries can be passed, because the journal receiver that was attached when the receive journal entry operation started is no longer attached.

Note: The system ends the RCVJRNE command after calling the exit program once with a reason code of 3.

  4 No journal entry is passed on this call to the exit program, and no more entries can be passed unless the local or remote journal is activated.

Note: This value can only be passed to the exit program when receiving journal entries from the attached receiver of a local or remote journal and the journal state for the journal is currently *INACTIVE.

  Char(1)
  Passed to the System from the Exit Program

  8 Requests the command processing program to start passing one or more journal entries in a block. » If BLKLEN(*NONE) was not specified, then specifying 8 in the first byte of the second parameter will have no effect and the first 5 bytes of the first parameter will be ignored. «

  9 Requests the RCVJRNE command to end. The exit program returns control to the system.

- Information in the second byte of the second parameter:

  Char(1)
  Passed to the Exit Program

N Additional journal entries are not currently available to be passed after this call of the exit program, or the RCVJRNE command will be ending after this call of the exit program.

Y Additional journal entries are currently available to be passed after this call of the exit program.

Any information passed from the exit program to the system in the second byte will be ignored.

The second byte of the second exit program parameter is provided whether journal entries are being passed as a single journal entry per call of the exit program, or as a block of journal entries per call.

- Information in the third byte of the second parameter:

Char(1)
Passed to the Exit Program

’00’x One or more journal entries are being passed to the exit program and the object names in the fixed length portion of each journal entry do not necessarily reflect the name of the object at the time the journal entry was deposited into the journal.

Note: This value is returned only when receiving journal entries from a journal receiver that was attached to a journal prior to V4R2M0.

0 No journal entries are currently being passed, so the information normally returned in this byte is not applicable.

1 One or more journal entries are being passed to the exit program and the object names in the fixed length portion of each journal entry reflect the name of the object at the time the journal entry was deposited into the journal.

2 One or more journal entries are being passed to the exit program and the object names in the fixed length portion of each journal entry do not necessarily reflect the name of the object at the time the journal entry was deposited into the journal. The object name in the fixed length portion of the journal entry may be returned as a previously known name for the object prior to the journal entry being deposited into the journal or be returned as *UNKNOWN.

Note: This value will only be returned when receiving journal entries from a remote journal and the remote journal is currently being caught up from its source journal. A remote journal is being caught up from its source journal when the Change Remote Journal (CHGRMTJRN) command or Change Journal State (QjoChangeJournalState) API is invoked and is currently replicating journal entries to the remote journal. After the call to the CHGRMTJRN command or QjoChangeJournalState API returns, the remote journal is maintained with a synchronous or asynchronous delivery mode, and the remote journal is no longer being caught up.

Any information passed from the exit program to the system in the third byte will be ignored.

The second byte of the second exit program parameter is provided whether journal entries are being passed as a single journal entry per call of the exit program, or as a block of journal entries per call.

Note: When an **N** is passed to the exit program in the second byte of the second parameter indicating that no additional journal entries are currently available, it does not necessarily mean that when the exit program returns, that the RCVJRNE command will have to wait for additional journal entries to be deposited into the journal. By the time the exit program returns, additional journal entries may already be available and depending upon what was specified on the DELAY parameter, may or may not be immediately passed to the exit program. If DELAY(N) was specified the system will wait N seconds before passing the journal entries to the exit program. If DELAY(*NEXTENT) was specified, the journal entries will immediately be passed to the exit program.

The third byte of the second exit program parameter is provided whether journal entries are being processed as a single journal entry per call of the exit program, or as a block of journal entries per call. When returned for a block of journal entries, the attribute applies to the object names for all of the journal entries being returned in the block.

For more information on the exit program and these two parameters used to receive the journal entries, see the Journal management article in the Information Center.

Optional Parameters

FILE Specifies a maximum of 300 qualified file names whose journal entries are received. This parameter also specifies the name of the file member whose journal entries are to be received.

To determine which journal entries are to be received, based on the specified file member name, the following is done:

- If the journal is a local journal, and if the specified file member currently exists on the system, the journal identifier is determined from the specified file member. All journal entries in the specified receiver range for that journal identifier are received.
- If the journal is a remote journal, or if the specified file member does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified file member. All journal entries in the specified receiver range for those journal identifiers are received.

  There can be more than one journal identifier associated with the specified file member if, for example, a file member was created by that name, it was journaled, and then deleted. Then another file member was created with the same name, and it was also journaled and then deleted. All of these actions would have to occur within the specified receiver range.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved, or restored. See the Journal management article in the Information Center for more information.
2. When specifying a database file on this parameter, journal entries with the following journal code values are received only if they satisfy the values specified on the other parameters:
   - Journal code D (database file-level information entries).
   - Journal code F (file member-level information entries).
   - Journal code R (record-level information entries).
   - Journal code U (user-generated entries).

- Other journal codes, if *IGNFILSLT is specified on that journal code. If *ALLSLT is specified on that journal code, no journal entries with that code are received.

***ALLFILE:** The search for the journal entries received is not limited to a specified file name.

Element 1: Physical File Name

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

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

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

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

***ALL:** Journal entries for all physical files in the specified library (the library name must be specified) whose journaled changes are currently in the journal receiver are received. If *ALL is specified and the user does not have the required authority to all of the files, an error occurs, and the command ends.

physical-file-name: Specify the name of the physical database file for which a journal entry is received.

Element 2: Member Name

***FIRST:** Entries for the first member in the file are received.

***ALL:** Entries for currently existing members of the file are received.

member-name: Specify the name of the member for which journal entries are received.

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

If the specified physical file does not exist on the system, specify either *ALL or a specific file member name.

RCVRNG

Specifies the starting (first) and ending (last) journal receivers used in the search for the journal entries that are received. The system starts the search with the starting journal receiver (as specified by the first value) and proceeds through the receiver chain until the ending journal receiver (as specified by the last value) is processed.

If dual receivers (receivers attached and detached in pairs) are used at any time, the system uses the first of the paired receivers when chaining through the receivers. The Work with Journal Attributes (WRKJRNA) command can be used to display the order of the receivers in the receiver chain.

If a problem is found in the receiver chain (such as damaged or not-found receivers) before the search operation begins, the system tries to use the second of the dual receivers. If these receivers also are damaged or not found, the operation ends.

***CURRENT:** The journal receiver that is currently attached when starting to receive journal entries is used.

***CURCHAIN:** The journal receiver chain that includes the journal receiver that is currently attached when starting to receive 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 receive journal entries.

Element 1: Starting Journal Receiver

The name of the journal receiver can be qualified by one of the following library values:

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

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

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

starting-journal-receiver: Specify the name of the first journal receiver that contains the journal entries to be received.

Element 2: Ending Journal Receiver

***CURRENT:** The journal receiver that is currently attached when starting to receive journal entries is used.

The name of the journal receiver can be qualified by one of the following library values:

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

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

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

ending-journal-receiver: Specify the qualified name of the last journal receiver that contains journal entries to be received. If the end of the receiver chain is reached before a receiver with this name is found, an error message is sent and no journal entries are received.

Note: The maximum number of receivers in the range is 256. If more receivers than this maximum are specified, an exception is signaled, and no journal entries are received.

FROMENT

Specifies the first journal entry considered for reception.

***FIRST:** The first journal entry in the journal receiver range specified is the first entry considered for reception.

starting-sequence-number: Specify the sequence number of the first journal entry considered for reception.

FROMTIME

Specifies the date and time of the first journal entry considered for reception. The journal entry with the specified date and time or the next later journal entry is the starting point for reception of journal entries.

Element 1: Starting Date

starting-date: Specify the date of the first journal entry considered for reception. The format of the date must be as defined by the job attributes DATFMT and, if separators are used, DATSEP.

Element 2: Starting Time

starting-time: Specify the time of the first journal entry considered for reception. The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

TOENT

Specifies the last journal entry considered for reception.

***NONE:** No journal entry is specified. Journal entries are passed to the exit program until the command is canceled (cancel request or cancel job) or until an end reason code (9) is set by the exit program. If there are no more entries to pass, the RCVJRNE command waits the number of seconds indicated on the DELAY parameter before trying to find more entries to pass.

***LAST:** The last journal entry in the journal receiver range specified is the last journal entry considered for reception.

ending-sequence-number: Specify the sequence number of the final journal entry considered for reception.

Note: The values specified for the from and to parameters can be the same. For example, FROMENT(234) and TOENT(234) can be specified.

TOTIME

Specifies the date and time of the last journal entry considered for reception. The first journal entry at or before the specified date and time is the last journal entry considered for reception.

Element 1: Ending Date

ending-date: Specify the date of the last journal entry considered for reception. The format of the date must be as defined by the job attributes DATFMT and, if separators are used, DATSEP.

Element 2: Ending Time

ending-time: Specify the time of the last journal entry considered for reception. See the FROMTIME parameter for a description of time formats.

NBRENT

Specifies the total number of journal entries to receive.

***ALL:** All journal entries that are for the specified journal receivers and that satisfy the selection values are received.

value: Specify the maximum number of journal entries to receive. If the specified journal entry identified by the TOENT or TOTIME parameter is reached before the value specified for NBRENT is met, the command ends normally.

JRNCDE

Specifies whether the entries being considered are limited to the journal entries that contain the specified journal code.

***ALL:** The journal entries received are not limited to those containing a specified code.

***CTL:** The journal entries received are those written to control the journal functions. These journal entries have codes J or F.

Element 1: Journal Code Value

journal-code: Specify the journal code to which journal entries are limited. Only journal entries with the specified journal code are received. A list of journal codes that can be specified is provided in the Journal management article in the Information Center.

Element 2: Journal Code Selection

***ALLSLT:** The journal entries with the specified journal code are received only if all other selection parameters are satisfied.

***IGNFILSLT:** The journal entries having the specified journal code are received only if all selection parameters, except the FILE parameter, are satisfied.

Note: This value is not valid for journal code D, F, or R.

ENTTYP

Specifies whether to limit the journal entries received to those of a specified journal entry type.

***ALL:** The journal entries to be received are not limited to a specified entry type.

***RCD:** The journal entries received are limited to those written for record level operations. These are entry types: BR, DL, DR, IL, PT, PX, UB, UP, and UR.

entry-type: Specify the entry type that limits the journal entries received. Only journal entries that contain the specified entry type are received. Up to 300 valid entry types can be specified. A list of valid entry types that can be received is in the Journal management article in the Information Center.

JOB Specifies whether the journal entries received are limited to those for a specified job. If the fully qualified name of the job is not specified, all journal entries that contain the simple job name are considered for reception.

A job identifier is a special value or a qualified name with up to three elements. For example:

```
*ALL
*
job-name
user-name/job-name
job-number/user-name/job-name
```

More information on this parameter is in Commonly used parameters.

***ALL:** The journal entries received are not limited to a specified job.

***:** The journal entries received are limited to those for the current job.

job-name: Specify the name of the job whose journal entries are considered for reception.

user-name: Specify the name of the user of the job whose journal entries are considered for reception.

job-number: Specify the number of the job whose journal entries are considered for reception.

PGM Specifies whether the journal entries received are limited to those created for a specified program.

***ALL:** The journal entries received are not limited to those created by a specified program.

program-name: Specify the name of the program whose journal entries are considered for reception. Only journal entries for this program are considered for reception.

USRPRF

Specifies that the journal entries to be received are limited to the journal entries for a specified user profile name.

***ALL:** The journal entries received are not limited to entries for a specified user profile.

user-name: Specify the name of the user profile whose journal entries are considered for reception. Only journal entries for this user profile are considered for reception.

CMTCYCID

Specifies that the journal entries are limited to the journal entries that contain the specified commit cycle identifier.

***ALL:** The journal entries received are not limited to a specified commit cycle identifier.

commit-cycle-identifier: Specify the commit cycle identifier that limits the journal entries received. Only journal entries that contain the commit cycle ID are considered for reception.

DEPENT

Specifies whether to receive the journal entries recording actions

- that occur as a result of a trigger program
- on records that are part of a referential constraint
- that will be ignored during an Apply Journaled Changes (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 received.

***NONE:** The journal entries relating to trigger programs, referential constraints and the entries which will be ignored by an Apply or Remove Journaled Changes operations are not received.

DELAY

Specifies the number of seconds that the command processing program (CPP) waits for a new journal entry to arrive if the last entry has already been received. After the last entry in the journal is received and passed to the exit program, the CPP tries to receive the next entry. If no new journal entry exists, the exit program is passed a value of 0 in the first byte of the second parameter.

Note: This parameter is valid only when TOENT(*NONE) is specified, and the last receiver specified on the RCVRNG parameter identifies the journal receiver that is currently attached when journal entries are starting to be received.

When the last entry on the journal has been passed to the exit program and no journal entries are currently available to be passed to the exit program, one of the following occurs:

- If a number of seconds is specified for the first element in the list, the exit program is immediately called and a ’0’ is passed to the first byte of the second exit parameter indicating that no additional journal entries are currently available. When the exit program returns control to the command, the system delays for the specified number of seconds.

  When the delay time has expired, the system then checks whether any additional journal entries are available to be passed to the exit program. Any additional entries are passed to the exit program sequentially, until there are no more available. When there are no further journal entries available, the exit program is called, and a ’0’ is passed as the first byte of the second exit program parameter, indicating there are no more journal entries currently available. When the exit program returns control to the command, the system again delays for the specified number of seconds.

If there are no new journal entries to pass to the exit program after the delay, the exit program is called, and a '0' is passed as the first byte of the second exit program parameter to indicate that no further journal entries are available. The exit program then can pass the value '9' for the first byte of the second parameter, indicating that this command is to end.

- If *NEXTENT is specified for the first element in the list, then additional journal entries are passed to the exit program as they become available. When this option is used, the second element in the list indicates the maximum number of seconds between calls to the exit program. If there are no additional journal entries to pass after the specified maximum delay time, the exit program is called, and a '0' is passed to the first byte of the second exit program parameter, indicating that no additional journal entries are currently available.

  The maximum delay time can be either of the following:

  – The time between a call to the exit program passing the last currently available journal entry, and a subsequent call to the exit program indicating that no new journal entries are available.

  – The time between calls to the exit program indicating that no additional journal entries are available.

  If the exit program is called after the maximum delay has expired, it then can pass the value '9' for the first byte of the second parameter, indicating that this command is to end.

Note: The previous description of the DELAY parameter assumes that the journal receiver that is currently attached at the beginning of the RCVJRNE command is still attached. If that journal receiver has been detached, the exit program is sent the reason code 3 after all journal entries have been received by the exit program and the RCVJRNE command ends.

30: The command delays 30 seconds before checking whether additional journal entries are available to be passed to the exit program.

seconds: Specify the length, in seconds, that the command delays before checking whether additional journal entries are available to be passed to the exit program. Valid values range from 1 through 99999.

Element 1: Delay Time Value

***NEXTENT:** A fixed delay time is not used. Additional journal entries are passed to the exit program as they become available.

Note: If the RCVJRNE exit program causes any additional calls of the RCVJRNE command, those additional calls cannot specify DELAY(*NEXTENT) if a preceeding call specified TOENT(*NONE).

Element 2: Maximum Delay Time

This element indicates the maximum number of seconds between calls to the exit program when a fixed delay time is not specified on the first element. This element is valid only if *NEXTENT is specified for the first element.

***CLS:** The process default wait time is used as the maximum number of seconds between calls to the exit program.

seconds: Specify the maximum length of time between calls to the exit program, in seconds. Valid values range from 1 to 99999.

ENTFMT

Specifies the format of the journal entries being received. For a description of what is represented by each of the fields in the journal entry, see the Journal management article in the Information Center.

***TYPE1:** The journal entries received are formatted to include the minimum information that can be specified. The information fields and the format of the information in each journal entry is shown below:

Table 1. *TYPE1 Journal Entry Format

Field Name	Length	From	To
Entry Length	5[1]	1	5
Sequence Number	10	6	15
Journal Code	1	16	16
Journal Entry Type	2	17	18
Date	6	19	24
Time	6	25	30
Job Name	10	31	40
User Name	10	41	50
Job Number	6	51	56
Program Name	10	57	66
Object Name	10	67	76
Object Library	10	77	86
Member Name	10	87	96
Count/RRN	10	97	106
Flag	1	107	107
Commit Cycle ID	10	108	117
Incomplete data	1	118	118
Minimized Entry Specific Data	1	119	119
Reserved	6	120	125
Entry-Specific Data	N[2]	126	N+125

Note:

[1] The length of the entry will be in bytes if BLKLEN(*NONE) was specified or in kilobytes if a value other than BLKLEN(*NONE) was specified.

[2] The length of the entry-specific data field varies from entry to entry. It is long enough to accommodate all the entry-specific data in each received journal entry.

***TYPE2:** The journal entries received include the information returned when ENTFMT(*TYPE1) is specified, the user profile field, which gives the name of the user who caused the logging of the received journal entries, and the name of the system on which the entry was sent.

The format for *TYPE2 journal entries is shown below.

Table 2. *TYPE2 Journal Entry Format

Field Name	Length	From	To
Entry Length	5[1]	1	5
Sequence Number	10	6	15
Journal Code	1	16	16

Field Name	Length	From	To
Journal Entry Type	2	17	18
Date	6	19	24
Time	6	25	30
Job Name	10	31	40
User Name	10	41	50
Job Number	6	51	56
Program Name	10	57	66
Object Name	10	67	76
Object Library	10	77	86
Member Name	10	87	96
Count/RRN	10	97	106
Flag	1	107	107
Commit Cycle ID	10	108	117
User Profile	10	118	127
System Name	8	128	135
Incomplete data	1	136	136
Minimized Entry Specific Data	1	137	137
Reserved	18	138	155
Entry-Specific Data	N[2]	156	N + 155

Note:

[1] » The length of the entry will be in bytes if BLKLEN(*NONE) was specified or in kilobytes if a value other than BLKLEN(*NONE) was specified. «

[2] The length of the entry-specific data field varies from entry to entry. It is long enough to accommodate all the entry-specific data in each received journal entry.

***TYPE3:** The journal entries received include the information returned when ENTFMT(*TYPE2) is specified, and the null value indicators. The format of the received entries depends on the value specified on the NULLINDLEN parameter. The tables in the NULLINDLEN parameter description show the three formats for *TYPE3.

***TYPE4:** The journal entries received include the information returned when ENTFMT(*TYPE3) is specified, the journal identifier, the physical file trigger indicator, and the referential constraint indicator. The format of the received entries depends on the value specified on the NULLINDLEN parameter. The tables in the NULLINDLEN parameter description show the three formats for *TYPE4.

***TYPEPTR:** The journal entries received include the information returned when ENTFMT(*TYPE4) is specified, and the entry specific data could contain pointers for specific journal entry types. The format of the received entries depends on the value specified on the NULLINDLEN parameter. The format of the received entries matches » format RJNE0100 which is « described in the Retrieve Journal Entries (QjoRetrieveJournalEntries) API. See the Application Program Interfaces (APIs) topic in the Information Center for the details associated with this format.

» ***JRNENTFMT:** The format of the received journal entries is determined by the JRNENTFMT parameter. The RTNPTR parameter indicates if the received journal entries could contain pointers. «

NULLINDLEN

Specifies the length, in bytes, used for the null value indicators portion of the journal entry received by the user. This parameter » is not valid if ENTFMT(*TYPE1) or ENTFMT(*TYPE2) « is specified.

***ENTFMT:** All null value indicators are received for each journal entry. The format for this value is shown in the following tables.

Note: The number of null value indicators, as well as the length of the entry-specific data can vary from entry to entry. In the following table, the number of null value indicators is designated by the variable 'M' and the length of the entry-specific data is designated by the variable 'N'.

Note: If the users selects the *TYPE3 format, the following information is not available in this format:

1. Incomplete Data indicating if the journal entry data is incomplete due to either LOB fields or Byte Stream File write operations.
2. Minimized Entry Specific Data indicating if the journal entry has minimized entry specific data because the journal had MINENTDTA specified for the object type of the journal entry.

See Journal management article in the Information Center for more information on the incomplete data indicator, the minimized entry specific data indicator, and these journal entries.

Table 3. NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*TYPE3)

Field Name	Length	From	To
Entry Length	5[1]	1	5
Sequence Number	10	6	15
Journal Code	1	16	16
Journal Entry Type	2	17	18
Timestamp	26	19	44
Job Name	10	45	54
User Name	10	55	64
Job Number	6	65	70
Program Name	10	71	80
Object Name	10	81	90
Object Library	10	91	100
Member Name	10	101	110
Count/RRN	10	111	120
Flag	1	121	121
Commit Cycle ID	10	122	131
User Profile	10	132	141
System Name	8	142	149
Number of Null Value Indicators[2]	5	150	154
Null Value Indicators	M	155	154+M
Length of Entry-Specific Data[3]	5	155+M	159+M
Entry-Specific Data	N	160+M	159+ M+N

Notes:

[1] » The length of the entry will be in bytes if BLKLEN(*NONE) was specified or in kilobytes if a value other than BLKLEN(*NONE) was specified. «

[2] This field contains the number of null value indicators (in decimal digits) in the received journal entry.

[3] This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.

Table 4. NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*TYPE4)

Field Name	Length	From	To
Entry Length	5[1]	1	5

Field Name	Length	From	To
Sequence Number	10	6	15
Journal Code	1	16	16
Journal Entry Type	2	17	18
Timestamp	26	19	44
Job Name	10	45	54
User Name	10	55	64
Job Number	6	65	70
Program Name	10	71	80
Object Name	10	81	90
Object Library	10	91	100
Member Name	10	101	110
Count/RRN	10	111	120
Flag	1	121	121
Commit Cycle ID	10	122	131
User Profile	10	132	141
System Name	8	142	149
Journal Identifier	10	150	159
Referential Constraint	1	160	160
Trigger	1	161	161
Incomplete data	1	162	162
Ignore during APYJRNCHG or RMVJRNCHG	1	163	163
Minimized Entry Specific Data	1	164	164
Reserved	5	165	169
Number of Null Value Indicators[2]	5	170	174
Null Value Indicators	M	175	174+M
Length of Entry-Specific Data[3]	5	175+M	179+M
Entry-Specific Data	N	180+M	179+ M+N

Notes:

[1] » The length of the entry will be in bytes if BLKLEN(*NONE) was specified or in kilobytes if a value other than BLKLEN(*NONE) was specified. «

[2] This field contains the number of null value indicators (in decimal digits) in the received journal entry.

[3] This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.

»

Table 5. NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*TYPEPTR) or ENTFMT(*JRNENTFMT)

For ENTFMT(*TYPEPTR), the format of this data is the same as RJNE0100 in the Retrieve Journal Entries (QjoRetrieveJournalEntries) API. For ENTFMT(*JRNENTFMT), the format of the data is determined by the JRNENTFMT parameter. « See the Application Program Interfaces (APIs) topic in the Information Center for the details associated with these formats.

field-length: Specify the field length for the null value indicators portion of the received journal entry. Valid values range from 1 to 8000 characters. The format of the received journal entry is shown in the following tables.

Table 6. NULLINDLEN(field-length) Journal Entry Format for ENTFMT(*TYPE3)

Field Name	Length	From	To
Entry Length	5[1]	1	5
Sequence Number	10	6	15
Journal Code	1	16	16
Journal Entry Type	2	17	18
Timestamp	26	19	44
Job Name	10	45	54
User Name	10	55	64
Job Number	6	65	70
Program Name	10	71	80
Object Name	10	81	90
Object Library	10	91	100
Member Name	10	101	110
Count/RRN	10	111	120
Flag	1	121	121
Commit Cycle ID	10	122	131
User Profile	10	132	141
System Name	8	142	149
Null Value Indicators	*field length*[2]	150	149+ *field length*
Entry-Specific Data	M[3]	150+ *field length*	149+ M+ *field length*

Notes:

[1] » The length of the entry will be in bytes if BLKLEN(*NONE) was specified or in kilobytes if a value other than BLKLEN(*NONE) was specified. «

[2] The length of the null value indicators field is the length specified on the NULLINDLEN parameter.

[3] The length of the entry-specific data field varies from entry to entry. It is long enough to accommodate all the entry-specific data in each received journal entry.

Table 7. NULLINDLEN(field-length) Journal Entry Format for ENTFMT(*TYPE4)

Field Name	Length	From	To
Entry Length	5[1]	1	5
Sequence Number	10	6	15
Journal Code	1	16	16
Journal Entry Type	2	17	18
Timestamp	26	19	44
Job Name	10	45	54
User Name	10	55	64
Job Number	6	65	70
Program Name	10	71	80
Object Name	10	81	90
Object Library	10	91	100
Member Name	10	101	110
Count/RRN	10	111	120
Flag	1	121	121
Commit Cycle ID	10	122	131
User Profile	10	132	141
System Name	8	142	149
Journal Identifier	10	150	159
Referential Constraint	1	160	160
Trigger	1	161	161
Incomplete data	1	162	162

Field Name	Length	From	To
Ignore during APYJRNCHG or RMVJRNCHG	1	163	163
Minimized Entry Specific Data	1	164	164
Reserved	5	165	169
Null Value Indicators	*field length*[2]	170	169+ *field length*
Entry-Specific Data	M[3]	170+ *field length*	169+ M+ *field length*

Notes:

[1] » The length of the entry will be in bytes if BLKLEN(*NONE) was specified or in kilobytes if a value other than BLKLEN(*NONE) was specified. «

[2] The length of the null value indicators field is the length specified on the NULLINDLEN parameter.

[3] The length of the entry-specific data field varies from entry to entry and is designated by the variable M. This length accommodates all of the entry-specific data in each received journal entry.

»

Table 8. NULLINDLEN(field-length) Journal Entry Format for ENTFMT(*TYPEPTR) or ENTFMT(*JRNENTFMT)

For ENTFMT(*TYPEPTR), the format of this data is the same as RJNE0100 in the Retrieve Journal Entries (QjoRetrieveJournalEntries) API. For ENTFMT(*JRNENTFMT), the format of the data is determined by the JRNENTFMT parameter. « See the Application Program Interfaces (APIs) topic in the Information Center for the details associated with these formats.

If the journal entry being passed to the exit program has fewer null value indicators than the length specified on the NULLINDLEN parameter, the trailing bytes are set to 'F0'X. Conversely, if a journal entry received has more null value indicators than the specified field length and truncation will result in the loss of a 'F1'X indicator value, the RCVJRNE request is ended.

Element 1: Variable-Length Field

***VARLEN:** The null value indicators field is a variable-length field. The received journal entry has the format specified in the table at the end of this parameter description.

Element 2: Maximum Field Length

The maximum length of the null value indicators field.

maximum-field-length: Specify the maximum number of null value indicators to be included in each received journal entry. Valid values range from 1 to 8000. If a journal entry received has more null value indicators than the specified field length and truncation will result in the loss of a 'F1'X indicator value, the RCVJRNE request is ended. The format of the received journal entry is shown in the following tables.

Table 9. NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTFMT(*TYPE3)

Field Name	Length	From	To
Entry Length	5[1]	1	5
Sequence Number	10	6	15
Journal Code	1	16	16
Journal Entry Type	2	17	18
Timestamp	26	19	44
Job Name	10	45	54

Field Name	Length	From	To
User Name	10	55	64
Job Number	6	65	70
Program Name	10	71	80
Object Name	10	81	90
Object Library	10	91	100
Member Name	10	101	110
Count/RRN	10	111	120
Flag	1	121	121
Commit Cycle ID	10	122	131
User Profile	10	132	141
System Name	8	142	149
Number of Null Value Indicators[2]	2	150	151
Null Value Indicators	*field length*	152	151+ *field length*
Length of Entry -Specific Data[3]	5	152+ *field length*	156+ *field length*
Entry-Specific Data	M	157+ *field length*	156+ M+ *field length*

Notes:

[1] » The length of the entry will be in bytes if BLKLEN(*NONE) was specified or in kilobytes if a value other than BLKLEN(*NONE) was specified. «

[2] This field contains the number of null value indicators (in binary digits) in the received journal entry.

[3] This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.

Table 10. NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTFMT(*TYPE4)

Field Name	Length	From	To
Entry Length	5[1]	1	5
Sequence Number	10	6	15
Journal Code	1	16	16
Journal Entry Type	2	17	18
Timestamp	26	19	44
Job Name	10	45	54
User Name	10	55	64
Job Number	6	65	70
Program Name	10	71	80
Object Name	10	81	90
Object Library	10	91	100
Member Name	10	101	110
Count/RRN	10	111	120
Flag	1	121	121
Commit Cycle ID	10	122	131
User Profile	10	132	141
System Name	8	142	149
Journal Identifier	10	150	159
Referential Constraint	1	160	160
Trigger	1	161	161
Incomplete data	1	162	162
Ignore during APYJRNCHG or RMVJRNCHG	1	163	163
Minimized Entry Specific Data	1	164	164
Reserved	5	165	169
Number of Null Value Indicators[2]	2	170	171
Null Value Indicators	*field length*	172	171+ *field length*
Length of Entry -Specific Data[3]	5	172+ *field length*	176+ *field length*

Field Name	Length	From	To
Entry-Specific Data	M[4]	177+ *field length*	176+ M+ *field length*

Notes:

[1] » The length of the entry will be in bytes if BLKLEN(*NONE) was specified or in kilobytes if a value other than BLKLEN(*NONE) was specified. «

[2] This field contains the number of null value indicators (in binary digits) in the received journal entry.

[3] This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.

[4] The length of entry-specific data can vary from entry to entry and is designated by the variable M.

»

Table 11. NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTFMT(*JRNENTFMT)

The format of this data is determined by the JRNENTFMT parameter. JRNENTFMT formats are described in the Retrieve Journal Entries (QjoRetrieveJournalEntries) API. See the Application Program Interfaces (APIs) topic in the Information Center for the details associated with these formats.

«

INCENT

Specifies whether only the confirmed or both the confirmed and unconfirmed, journal entries are received. This parameter only applies when receiving journal entries for output from a remote journal.

Confirmed entries are those journal entries which have been sent to this remote journal and the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is known.

Unconfirmed entries are those journal entries which have been sent to this remote journal, but the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is not known, or the object name information for those journal entries is not yet known to the remote journal. Unconfirmed journal entries can only exist within the attached receiver of a remote journal. This only applies if synchronous delivery mode is being used for a particular remote journal.

***CONFIRMED:** Only those journal entries which have been confirmed are received.

***ALL:** All confirmed and unconfirmed journal entries are received.

» JRNENTFMT

Specifies the format of the journal entries received by the exit program. The formats are described in the Retrieve Journal Entries (QjoRetrieveJournalEntries) API. See the Application Program Interfaces (APIs) topic in the Information Center for the details associated with these formats. This parameter is only valid if ENTFMT(*JRNENTFMT) is also specified.

RJNE0100: Received journal entries are in RJNE0100 format.

RJNE0200: Received journal entries are in RJNE0200 format.

RTNPTR

Specifies whether the journal entries received include entry specific data that could contain pointers for specific journal entry types. This parameter is only valid if ENTFMT(*JRNENTFMT) is also specified.

***NONE:** Received journal entries will not include pointers.

***SYSMNG:** Specifies that received journal entries received include entry specific data that could contain pointers for specific journal entry types. The system will manage the releasing of the resources related to pointers that are received in the journal entries. The pointers, their associated pointer handles and any related storage will be released when the exit program returns control to the system.

***USRMNG:** Specifies that received journal entries received include entry specific data that could contain pointers for specific journal entry types. The user will manage the releasing of the resources related to pointers that are received in the journal entries. Neither the pointers, their associated pointer handles or any related storage will be released when the exit program returns control to the system. The user must use the Delete Pointer Handle (QjoDeletePointerHandle) API to release these resources. See the Application Program Interfaces (APIs) topic in the Information Center for the details associated with this API.

BLKLEN

Specifies whether the system will be sending one or more journal entries to the exit program and specifies the block length of the buffer passed to the exit program. The EXITPGM parameter has further details about block mode semantics.

***NONE:** At most one journal entry will be sent to the exit program.

***CALC:** One or more journal entries will be passed to the exit program in a block. The length of the block passed (the first parameter passed to the exit program) is determined by the system and will be optimal.

block-length: Specify the length in kilobytes of the buffer passed to the exit program (EXITPGM parameter). Valid values range from 32 to 4000. «

Examples for RCVJRNE

Example 1: Receiving Journal Entries

```
RCVJRNE   JRN(APPLIB/JRN1)  EXITPGM(MYLIB/RCVPGM)
  FILE(APPLIB/FILE3)  TOENT(*LAST)
  ENTFMT(*TYPE3)  NULLINDLEN(*ENTFMT)
```

This command receives journal entries from the journal receiver that is currently attached (when journal entries are starting to be received) to the journal JRN1 in library APPLIB and passes them one at a time to program RCVPGM in library MYLIB. Only entries with file-level information for the first member of file FILE3 in library APPLIB are received. The format of each entry passed to the exit program is shown in the “NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*TYPE3)” table shown within the NULLINDLEN parameter description.

Example 2: Receiving Journal Entries

```
RCVJRNE   JRN(JRNLIB/MYJRN)  EXITPGM(RCVLIB/PGMA)
  FILE(FILELIB/PFILEB MBRONE)  TOENT(*LAST)
  ENTFMT(*TYPE3)  NULLINDLEN(*VARLEN 30)
```

This command receives journal entries with file-level information for member MBRONE of file PFILEB in library FILELIB from the journal receiver currently attached (when journal entries are starting to be received) to journal MYJRN in library JRNLIB and sends them one at a time to program PGMA in library RCVLIB. The format of each entry passed to the exit program is shown in the “NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTFMT(*TYPE3)” table at the end of the NULLINDLEN parameter description. The null value indicators portion of each received entry is 30 characters in length.

Example 3: Receiving Journal Entries Using DELAY(*NEXTENT)

```
RCVJRNE   JRN(JRNLIB/MYJRN)  EXITPGM(RCVLIB/PGMA)
  RCVRNG(*CURCHAIN)  TOENT(*NONE)  DELAY(*NEXTENT)
```

This command receives all available journal entries from the chain of journal receivers, which includes the journal receiver that is attached at the start of receiving journal entries, associated with the journal MYJRN in the library JRNLIB. These journal entries are sent sequentially to exit program PGMA in library RCVLIB, as they become available. The maximum length of time between calls to the exit program is equal to the process default wait time value.

Error messages for RCVJRNE

***ESCAPE Messages**

CPF7002
File &1 in library &2 not a physical file.

CPF7006
Member &3 not found in file &1 in &2.

CPF7007
Cannot allocate member &3 file &1 in &2.

CPF701B
Journal recovery of an interrupted operation failed.

CPF705C
INCENT(*ALL) not allowed for a local journal.

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

CPF7054
FROM and TO values not valid.

CPF7055
Maximum number of objects exceeded.

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

CPF706A
Significant null value indicator truncated.

CPF706D
RCVJRNE exit program &1 in &2 failed.

CPF7060
File &1 member &5 never journaled to journal &3.

CPF7061
Conversion of journal entries failed.

CPF7062
No entries converted or received from journal &1.

CPF7065
Entry type (ENTTYP) not valid for journal code (JRNCDE).

CPF707B
DELAY(*NEXTENT) not allowed.

CPF7074
RCVRNG for specified SEARCH not valid.

CPF708C
DELAY(*NEXTENT) not allowed with INCENT(*ALL).

CPF708D
: Journal receiver found logically damaged.

CPF709C
: JOB, PGM, and USRPRF not valid for receiver range.

CPF7096
: Ending receiver for RCVRNG must identify an attached receiver.

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.

RCVMSG (Receive Message) Command Description

RCVMSG Command syntax diagram

Purpose

The Receive Message (RCVMSG) command is used by a program to receive a message that was previously sent to a message queue.

The RCVMSG command receives messages from a job message queue (a message queue associated with a call stack entry or the external message queue (*EXT)), or from a named message queue. The program can receive a message from a message queue associated with its own call stack entry or from a message queue associated with another call stack entry.

This command copies a message received in the specified message queue into control language (CL) variables within the program. The message and its attributes are copied into the CL variables specified by the parameters KEYVAR through DTACCSID.

You can specify the message being received by indicating the message type, the reference key of the message, or both. The program receiving the message can also specify, on the RCVMSG command, whether a message is removed from the message queue or left there as an old message. If the specified message queue is not allocated to the job in which this command is entered, or to any other job, the message queue is implicitly allocated by this command for the duration of the command's processing.

If a message of the specified type has not been received by the queue, the requesting program can either wait for a message to arrive or continue with other processing. This allows a set of message queues to be polled.

If the message received is an unhandled exception message, the program can specify whether this command should handle the exception. An unhandled exception message is an escape, status, or notify message that has been sent to an Integrated Language Environment (ILE) procedure. When this command is run, the ILE procedure has not yet taken action to tell the system that the exception is handled. One action the ILE procedure can take is to call a CL program that receives the message using this command. More information on actions that can be taken is in the ILE C book.

Restriction: This command is valid only in compiled CL programs.

Optional Parameters or if it is associated with the caller of the program or procedure.

***SAME:** The message is received from the message queue of the program or procedure identified by Element 2.

***PRV:** The message is received from the message queue of the program or procedure that called the program or procedure identified by Element 2.

Note: If the message queue previous to the one identified by Element 2 is for an ILE program entry procedure (PEP), the message will be received from the message queue immediately previous to the PEP message queue. (Effectively this would be two message queues previous to the one identified by Element 2).

Element 2: Program or Qualified Procedure

Element 2 of this parameter has three items. Item 1 specifies the program or procedure of the job message queue. Items 2 and 3 specify the module name and the bound program name, which can be used to qualify the procedure name.

Item 1: Program or Procedure Name

***:** Identifies the program running the RCVMSG command.

program-or-procedure-name: Specify the name of the program or procedure used.

If Item 1 identifies a program, the name specified can be a maximum of 10 characters. If Item 1 identifies a procedure, the name specified can be a maximum of 256 characters.

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.

Item 2: Module Name

The module name qualifier identifies the module into which the procedure was compiled.

***NONE:** No module name is specified.

module-name: Specify the module name to be used as a qualifier for the specified procedure name (modules are associated only with procedures). The module name can be a maximum of 10 characters.

If a module name is not specified but a bound program name is, *NONE must be specified in the module name position.

Item 3: Bound Program Name

The bound program name qualifier identifies the program to which the procedure was bound.

***NONE:** No bound program name is specified.

bound-program-name: Specify the bound program name to be used as a qualifier for the specified procedure name (and module name if specified). The bound program name can be a maximum of 10 characters.

Single Value

***EXT:** The message is received from the external message queue of the job. The external message queue is used to communicate with the external requester of the job, such as a display station user.

MSGQ

Specifies the qualified name of the message queue (not a program message queue) from which a message is received. If MSGQ is specified, the PGMQ parameter cannot be specified.

***PGMQ:** The job message queue specified in the PGMQ parameter is the only queue from which a message is received. The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***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.

library-name: Specify the name of the library where the message queue is located.

message-queue-name: Specify the name of the message queue from which a message is received.

MSGTYPE

Specifies the type of message received by this program. The message types *EXCP, and *RQS exists only on job message queues and therefore can be received only from a job message queue. For the coding relationships between the MSGTYPE and MSGKEY parameters, see Coding Relationships in the MSGKEY parameter description.

***ANY:** Any type of message (except a sender's copy) is received. To receive a sender's message, MSGTYPE(*COPY) must be specified.

***COPY:** A copy of an inquiry message that was previously sent is received by this program. The message queue specified for the PGMQ or MSGQ parameters must be the same queue that was specified for the RPYMSGQ parameter when the inquiry message was sent.

***COMP:** A completion message is received. It indicates the status of the work that this program requested of another program.

***DIAG:** A diagnostic message is received. It provides information about errors detected by another program in the input sent by this program or errors that occurred when the requested function was being processed by the other program.

***EXCP:** An exception message is received. Exception messages (escape, notify, status) are received by the program in last-in first-out (LIFO) order. The receiving program can monitor for exception messages by using the MONMSG command.

Note: Non-exception messages are received in first-in first-out (FIFO) order.

If an exception message is received from a message queue for a procedure, the related exception may not be handled at the time the RCVMSG command is run. The RMV parameter can be used to specify whether the exception is to be handled by the RCVMSG command.

***FIRST:** The first message currently on the specified message queue is received.

***INFO:** An information only message is received.

***INQ:** An inquiry message is received.

***LAST:** The last message currently on the specified message queue is received.

***NEXT:** The message that follows the one specified in the MSGKEY parameter is received. If there is not another message available, blanks are returned in all CL variables.

When a message is received from a message queue associated with a call stack entry, *NEXT works only for one call stack entry. *NEXT cannot be used to receive messages for multiple call stack entries of the same program.

***PRV:** The message previous to the message indicated by the MSGKEY parameter is received.

***RPY:** A reply message is received. This program has sent an inquiry message to a message queue and expects a reply.

***RQS:** A request message is received. The message specifies a request for a function. The receiving program can then perform the function requested.

MSGKEY

Specifies the message reference key of the message that is received.

***NONE:** No message reference key is specified.

***TOP:** The top of the message queue is used. *TOP can be used only when MSGTYPE(*NEXT) is specified. It causes the first message on the message queue to be received. For program message queues, this is the message following the last request message that was received, if any.

&CL-variable-name: Specify the name of the control language (CL) variable that contains the message reference key of the message received by this program. This key is assigned by the system and cannot be shown. The variable must be a character variable having a length of 4 characters.

Coding Relationships: The MSGTYPE and MSGKEY parameters can be used separately in the RCVMSG command, or together.

- If neither MSGTYPE nor MSGKEY is specified, MSGTYPE(*ANY) is assumed and the first *new* message in the queue is received; that is, the messages are received in FIFO (first-in, first-out) order.
- If one of the message types specified on the MSGTYPE parameter is *COMP, *DIAG, *INFO, *INQ, *RPY, *COPY, or *RQS, a new message of the specified type is received in FIFO order. If the type is *EXCP, new messages are received in LIFO (last-in, first-out) order.
- If only MSGKEY is specified with a CL variable name and the message queue contains a message with the specified message reference key, that message is received. If the MSGKEY specified is for a sender's copy message, the reply to the message is received if it is available. If the reply is not available, blanks will be returned in all CL variables. If a message is requested by key and the message is not available, an escape message is sent to the requesting program.
- If MSGTYPE(*COPY) and MSGKEY (&CL-variable-name) are specified, the sender's copy of an inquiry message is received.

- If both MSGTYPE and MSGKEY (&CL-variable-name) are specified and the message queue has a message of that type, the message is received by the program. If the reference key is correct and the message type is not, then an error message is sent to the program.
- If MSGTYPE(*NEXT) is specified, MSGKEY must be specified. The message following the message with the specified reference key is received. If MSGKEY(*TOP) is specified with MSGTYPE(*NEXT), the first message on the message queue is received. For call message queues, this is the first message following the last request message received.
- If MSGTYPE(*PRV) is specified, MSGKEY must be specified. The message previous to the message with the specified reference key is recieved.

If MSGTYPE(*RPY) is specified with a MSGKEY value that refers to either a sender's copy or an inquiry message, any reply to the sender's copy or inquiry message is returned. If there is no reply to that sender's copy or inquiry message, blanks are returned. When the MSGKEY value refers to an inquiry message, the WAIT parameter is ignored (this implies WAIT(0), which is the default value for the WAIT parameter).

If MSGTYPE(*ANY) is specified with a KEYVAR variable and the first message type found is a reply message, the KEYVAR variable returns the message reference key of the sender's copy message. Similarly, if MSGTYPE(*RPY) is specified with a KEYVAR variable, the message reference key of the sender's copy message is returned.

WAIT Specifies the length of time (in seconds) that the program waits for a message of the specified type to arrive in the message queue if it is not there when this RCVMSG command is processed. If the message does not arrive in the specified time, the CL variables named to receive message fields are filled with blanks.

The program cannot wait for a message from a *program* message queue unless it is receiving a reply.

If a wait time is specified (not zero), the message queue is implicitly allocated to the first user whose message is received, and it is not released until the request has been handled by the program.

If a message is sent to a message queue in the same job, and the message queue is in break delivery mode, this parameter is ignored. (that implies WAIT(0), which is the default value for the WAIT parameter).

If the value specified for MSGKEY refers to an inquiry message, and MSGTYPE(*RPY) has been specified, the program ignores the WAIT parameter (Value for Wait is 0).

0: The program does not wait for the arrival of a message. If a message of the specified type is not in the queue when this command is processed, the specified CL variables are filled with blanks (or zeros, if they are decimal variables).

***MAX:** The program waits indefinitely for the arrival of the specified type of message.

number-of-seconds: Specifies the number of seconds that the program waits for the arrival of a specific type of message.

RMV Specifies whether the message received by the program is removed from the message queue. For exception messages for unhandled exceptions, also specifies whether the exception is to be handled by the RCVMSG command. If MSGTYPE(*INQ) is specified, RMV(*NO) must also be specified so a reply to the inquiry message can be sent.

***YES:** The message is removed from the message queue. If the message is an unhandled exception, the exception is handled by running the RCVMSG command.

***NO:** The message is not removed from the message queue. It is left on the message queue as an old message. If the message is an unhandled exception, the exception is handled by running the RCVMSG command.

Note: Old messages are messages that have been received but not deleted. An old message can be received again in one of the following ways:

1. The message reference key of the message is specified for the MSGKEY parameter.
2. A message type of *FIRST, *LAST, *NEXT, or *PRV is specified for the MSGTYPE parameter.

***KEEPEXCP:** If the message is an exception message and the exception has not been handled, the exception is left unhandled and the message is left on the message queue as a new message. It can be received again by using the RCVMSG command to receive an *EXCP message. If the message is not an exception message, or if it is but the exception has already been handled, the message is left on the message queue as an old message.

To handle an exception after the RCVMSG has been run, the command can be run a second time by specifying RMV(*YES) or RMV(*NO).

CCSID

Specifies the coded character set identifier (CCSID) that you want the message text returned in. This only applies to text returned in the MSG, SECLVL and MSGDTA parameters. When replacement data is returned in the MSGDTA parameter or substituted into the text returned in the MSG or SECLVL parameters, only the part of the replacement text that is defined as a character that can be converted (*CCHAR) is converted. The rest of the replacement data is not converted. For more information about the *CCHAR field, see the ADDMSGD command.

***JOB:** The received message is converted to the CCSID of the job before being returned.

***HEX:** The received message is not converted before being returned.

coded-character-set-identifier: Specify the CCSID that you want your message converted to before being returned. Valid values range from 1 through 65535. For a list of valid CCSID values, see the Globalization topic in the Information Center. Only CCSID values that a job can be changed to will be accepted.

For more information on the message handler and its use of CCSIDs, see the Globalization topic in the Information Center.

Parameters for Received Message Fields

All of the following parameters are used to specify the names of the CL variables that receive the specified fields and attributes of a message when the message is received by the program. If WAIT (number-of-seconds) is specified, and the time-out occurs, the variables, which must already be declared in the program, are filled with blanks. If the message field returned is larger than the CL variable specified, the message field is truncated; if the message field is shorter, it is padded with blanks. If the program does not need the value for a specific message parameter, no CL variable is specified for it. If a parameter is not specified, the corresponding message value is not received in the program.

KEYVAR

Specifies the name of the CL character variable, if any, that contains the message reference key identifying the message received by the program containing this RCVMSG command. At the time the RCVMSG command is processed, the system returns the message reference key to the variable specified by KEYVAR in this command and changes the received message to an old message. The message reference key can then be used in the MSGKEY parameter in a subsequent RCVMSG command to receive the old message. If the message is not found, blanks are returned for the KEYVAR variable. For reply type messages, use the MSGKEY parameter on this command in conjunction with the KEYVAR parameter on the SNDPGMMSG command. The message reference key can also be used by this program for building message subfiles. The CL variable is the name of the field for which the SFLMSGKEY keyword is specified in the DDS for the message subfile.

Note: For message queues not associated with call stack entries, message reference keys can be used again after a message has been received and then removed (by specifying *YES on the RMV parameter).

The variable must be a character variable having a length of 4 characters.

Note: When using the message reference key (obtained from the CL variable specified by the KEYVAR parameter of the Send Program Message (SNDPGMMSG) command) to receive the reply to an inquiry message, note that the message reference key refers to the sender's copy. The sender's copy message is located on the reply message queue (which defaults to the program message queue that sent the inquiry message), not the message queue to which the inquiry message was sent.

MSG
Specifies the name of the CL character variable, if any, that contains the message text when the message is received by the program. This includes the message data fields that were substituted for substitution variables in the text before the message was sent (replies and immediate messages contain no message data fields). This is a variable-length field, but most first-level message text is less than 132 characters in length. This parameter receives the reply from an *INQ message sent by the same program that issued the RCVMSG command.

MSGLEN
Specifies the name of the CL decimal variable, if any, that contains the total length of the first-level message text available to be received. The variable must be a decimal variable having a length of 5 positions.

SECLVL
Specifies the name of the CL character variable, if any, that contains the help text when the message is received by the program. This includes the message data fields that were substituted for substitution variables in the text before the message was sent (replies and immediate messages do not have second-level messages). A message data field is a variable length field, but most help text is less than 3000 characters in length.

SECLVLLEN
Specifies the name of the CL decimal variable, if any, that contains the total length of the second-level message available to be received. The variable must be a decimal variable having a length of 5 positions.

MSGDTA
Specifies the name of the CL character variable, if any, that contains the message data record received by the program as part of the message. The message data record contains the substitution values (in a single character string) that are used in the text of the received message. The amount of data returned and its format depend on the message. Pointers contained in system messages are invalidated.

Note: If you use data that has an invalidated pointer in it an error message can occur.

MSGDTALEN
Specifies the name of the CL decimal variable, if any, that contains the total length of the message data record available to be received. The variable must be a decimal variable having a length of 5 positions.

MSGID
Specifies the name of the CL character variable, if any, that contains the message identifier of the message received by the program. If the message being received is an immediate message, a message identifier is not returned. The minimum length of the variable is 7 characters.

SEV Specifies the name of the CL decimal variable, if any, that contains the severity code of the message received by the program. The variable must be a decimal variable having a length of two positions. If the message being received is an immediate message, the message severity is not returned.

SENDER
Specifies the name of the CL character variable, if any, that contains the identification of the sender of the message received through the RCVMSG command. The length of the CL variable depends on the SENDERFMT specification. If SENDERFMT(*SHORT) is specified, the variable should be a minimum of 80 characters. If SENDERFMT(*LONG) is specified, the variable should be a minimum of 720 characters.

SENDERFMT
Specifies which format of the sender identification is returned. This parameter is valid only when the SENDER parameter is specified.

***SHORT:** The short format of the sender information is returned. The short format is 80 characters, with the last 9 characters set to blanks. The following information is returned:

- The first 26 characters identify the sending job
  - Job name (10)
  - User name (10)
  - Job number (6)
- The next 16 characters identify the sending program
  - Program name (12) (for an ILE procedure, this is the bound program name); if the sender type is 3, the first three characters of this field are less than symbols (<<<) followed by the last nine characters of the program name
  - Instruction number (4) (for an ILE procedure, this field is set to blanks)
- The next 13 characters are the date and time
  - Date (7) (in the format 0yymmdd)
  - Time (6) (in the format hhmmss)
- The next 14 characters identify the sent-to call stack entry if the message is sent to a call message queue
  - Program name (10) (for an ILE procedure, this is the bound program name)
  - Instruction number (4) (for an ILE procedure, this field is set to blanks)
- The next 1 character identifies the sender type
  - "0*" if the sender is an OPM program or a SLIC program with 12 characters or less
  - "1" if the sender is an ILE procedure and the name is 256 characters or less
  - "2" if the sender is an ILE procedure and the name is more than 256 characters
  - "3" if the sender is a SLIC program with more than 12 characters
- The last 1 character identifies the sent-to type
  - "0" if the receiver is an OPM program
  - "1" if the receiver is an ILE procedure and the name is 256 characters or less
  - "2" if the receiver is an ILE procedure and the name is more than 256 characters

***LONG:** The long format of the sender information is returned. The long format is 720 characters, with the last 46 characters set to blanks. The following information is returned:

- The first 26 characters identify the sending job
  - Job name (10)
  - User name (10)
  - Job number (6)
- The next 13 characters are the date and time
  - Date (7) (in the format 0yymmdd)
  - Time (6) (in the format hhmmss)
- The next 1 character identifies the sender type
  - "0" if the sender is an OPM program or a SLIC program with 12 characters or less
  - "1" if the sender is an ILE procedure and the name is 256 characters or less
  - "2" if the sender is an ILE procedure and the name is more than 256 characters
  - "3" if the sender is a SLIC program with more than 12 characters
- The next 1 character identifies the sent-to type
  - "0" if the receiver is an OPM program
  - "1" if the receiver is an ILE procedure and the name is 256 characters or less
  - "2" if the receiver is an ILE procedure and the name is more than 256 characters.
- The next 12 characters are the sender's program name (for an ILE procedure, this is the bound program name); if the sender type is 3 and the program name is greater than 12 characters in length, the first three characters of this field are less than symbols (<<<) followed by the last nine characters of the program name.
- The next 10 characters are the sender's module name (if the sender is not an ILE procedure, this field is set to blanks)
- The next 256 characters are the sender's procedure name (if the sender is not an ILE procedure, this field is set to blanks)
  - For a nested procedure name, each procedure name is separated by a colon (:) starting with the outer-most procedure name , and ending with the inner-most procedure name.
  - For a procedure name that is longer than 256 characters, three less than symbols (<<<) are returned followed by the last 253 characters of the procedure name; the QMHRCPM API can be used to obtain the entire procedure name
- The next 1 character is blank
- The next 4 characters are the number of statement numbers available

Note: A statement number represents a point in the sending program at which the message was sent. For programs and non-optimized procedures, this count is always 1. For optimized procedures, this count can be greater than 1, and each statement number represents a point at which the message could have been sent. If it is not possible to return statement numbers, this count will be 0.

- The next 30 characters return a maximum of 3 statement numbers, 10 characters each
- The next 320 characters return program or procedure information if the message being received was originally sent to a message queue associated with a call stack entry (otherwise, this field is set to blanks)
  - Sent-to program name (10) (for an ILE procedure, this is the bound program name)
  - Sent-to module name (10) (if the sender is not an ILE procedure, this field is set to blanks)
  - Sent-to procedure name (256) (if the sender is not an ILE procedure, this field is set to blanks)

    - For a nested procedure name, each procedure name is separated by a colon (:) starting with the outer-most procedure name, and ending with the inner-most procedure name.
    - For a procedure name that is longer than 256 characters, three less than symbols (<<<) are returned followed by the last 253 characters of the procedure name; the QMHRCVPM API can be used to obtain the entire procedure name
  - Blanks (10)
  - Number of statements available for the receiving call stack entry (4)

Note: A statement number represents a point at which the sent-to program was suspended (for example, due to a call operation) at the time the message was sent. For programs and non-optimized procedures, this count is always 1. For optimized procedures, this count can be greater than 1, and each statement number represents a point at which the message could have been sent. If it is not possible to return statement numbers, this count will be 0.

  - Statement numbers (30) (a maximum of 3 statement numbers, 10 characters each)

RTNTYPE
Specifies the name of the CL variable, if any, that contains the type code for the message received by the program. The variable must be a character variable having a length of 2 positions. The following values are returned to indicate the message type:

Value	Message Type
01	Completion
02	Diagnostic
04	Information
05	Inquiry
06	Sender's Copy
08	Request
10	Request with prompting
14	Notify (exception already handled at time of RCVMSG)
15	Escape (exception already handled at time of RCVMSG)
16	Notify (exception not handled at time of RCVMSG)
17	Escape (exception not handled at time of RCVMSG)
21	Reply, not checked for validity
22	Reply, checked for validity
23	Reply, message default used
24	Reply, system default used
25	Reply, from System Reply List

ALROPT
Specifies the name of the CL variable, if any, used to return the alert option of the message received by the program. The variable must be a character variable 9 positions in length.

MSGF Specifies the name of the CL variable, if any, used to return the message file name of the message received by the program. If the message received is a stored message, the message file name of the file containing the stored message is returned. If the received message is not a stored message, the message file name is returned as blanks. The variable must be a character variable 10 positions in length.

Note: The message file name returned on this parameter is the message file specified or defaulted on either the SNDPGMMSG or SNDUSRMSG command, not the overriding message file. If an override was specified when sending the message, the same override should be used when receiving the message.

MSGFLIB
Specifies the name of the CL variable, if any, used to return the message file library name of the message received by the program. If the message received is a stored message, the message file library name specified when the message was sent is returned. If *LIBL was specified on the send command, *LIBL is returned. If the received message is not a stored message, the message file library name is returned as blanks. The variable must be a character variable 10 positions in length.

Note: The message file library name returned on this parameter is the message file specified or defaulted on either the SNDPGMMSG or SNDUSRMSG command, not the overriding message file library. If an override was specified when sending the message, the same override should be used when receiving the message.

SNDMSGFLIB
Specifies the name of the CL variable, if any, used to return the name of the library that contains the message file used to send the message. If the message received is stored in a message file, the name of the library containing the message file is returned. For example, if the name of the library containing the message file is MYLIB, and *LIBL is used to send a predefined message, this parameter would return MYLIB. If the message received is not a stored message, the message file library name is returned as blanks.

TXTCCSID
Specifies the name of the CL variable, if any, used to return the coded character set identifier (CCSID) associated with the text returned by the MSG and SECLVL parameters. If a conversion error occurs, or if the CCSID you requested the text to be converted to is 65535, the CCSID that the message description or the text for an immediate message is stored in is returned. Otherwise, the CCSID you wanted the text converted to is returned. If you do not want the text converted before it is returned to you but you do want to know which CCSID that the message description or the text for an immediate message is stored in, specify 65535 on the CCSID parameter, and the CCSID is returned by the TXTCCSID parameter. You can also check for a conversion error by comparing the CCSID you passed in against the TXTCCSID returned. If they are not equal and they are not 65535, a conversion error occurred.

DTACCSID
Specifies the name of the CL variable, if any, used to return the coded character set identifier (CCSID) associated with the replacement data defined as *CCHAR. All other replacement data is not converted before it is returned. If a conversion error occurs, or if the CCSID you requested the text to be converted to is 65535, the CCSID of the message data is returned. If there is no *CCHAR replacement data in the text, 65535 is returned. Otherwise, the CCSID you wanted the text converted to is returned. For immediate messages, 0 is returned. You can check for a conversion error by comparing the CCSID you passed in against the DTACCSID returned. If they are not equal and they are not 65535 or 0, a conversion error occurred.

Examples for RCVMSG

Example 1: Receiving a Message

```
RCVMSG  MSGQ(SMITH)  MSGKEY(&KEY)  MSG(&WORK)
```

This command receives the message having the message reference key specified by the program variable &KEY from the message queue SMITH. The text of the message is copied into the CL variable &WORK.

Example 2: Receiving a New Message

```
RCVMSG  MSGQ(INV)  WAIT(120)  MSG(&WORK)
```

This command receives a new message from the message queue named INV into the CL variable &WORK. The program waits no more than 120 seconds for the arrival of a new message if there are no new messages in the message queue. If there is more than one new message in the queue, the first message in the queue is the message received by the program.

Example 3: Receiving a Message From a Procedure

```
RCVMSG   PGMQ(*SAME CURRENT_MONTH_TOTALS)  MSGTYPE(*EXCP)
  RMV(*KEEPEXCP)  MSGID(&MID)  MSG(&MTEXT)
```

This command receives an exception message from the procedure CURRENT_MONTH_TOTALS. Since the specified name is more than 10 characters, the system does search for any programs. If the message is an unhandled exception, the message is left on the call message queue as a new message and the exception is not handled by the RCVMSG command. The message ID is returned in the CL variable &MID and the message text in the CL variable &MTEXT. To handle the exception and remove the message, run the following RCVMSG command:

```
RCVMSG   PGMQ(*SAME CURRENT_MONTH_TOTALS)
  MSGTYPE(*EXCP)  RMV(*YES)
```

Example 4: Receiving a Message from a Program or Procedure

```
RCVMSG   PGMQ(*SAME TARGETPGM)  MSGTYPE(*EXCP)
  RMV(*NO)  MSGID(&MID)  MSG(&MTEXT)
```

This command receives an exception message from the message queue of the program or procedure named TARGETPGM. Since the specified name is only 9 characters, the system searches both programs and procedures. Because RMV(*NO) is specified, if the message is an unhandled exception, the exception is handled by the RCVMSG command. The message is left on the message queue as an old message.

Example 5: Receiving a Message Using Qualifiers

```
RCVMSG   PGMQ(*SAME PRINT_RPT_FMT1 DEPTRPTS AREARPTS)
  MSGTYPE(*EXCP)  RMV(*YES)
  MSGID(&MID)  MSG(&MTEXT)
```

This command receives an exception message from the message queue of the procedure named PRINT_RPT_FMT1. The procedure must have been compiled into the module DEPTRPTS and have been bound into the bound program AREARPTS. Since RMV(*YES) is specified, the exception is handled if the exception message is for an unhandled exception. The message is always removed from the message queue.

Example 6: Receiving a Message Using a Partial Procedure Name

```
RCVMSG   PGMQ(*SAME HANDLE_FORM_NUM>>>)
  MSGID(&MID)  MSG(&MTEXT)
```

This command receives a new message from the most recent procedure whose name begins with HANDLE_FORM_NUM.

Error messages for RCVMSG

*ESCAPE Messages

CPF2401
Not authorized to library &1.

CPF2403
Message queue &1 in &2 not found.

CPF2407
Message file &1 in &2 not found.

CPF2408
Not authorized to message queue &1.

CPF2411
Not authorized to message file &1 in &2.

CPF241C
Variable for SENDER parameter is too small.

CPF2410
Message key not found in message queue &1.

CPF2415
End of requests.

CPF2423
Variable specified in SENDER parameter less than &1 bytes.

CPF2433
Function not allowed for system log message queue &1.

CPF2449
Message that should be a reply, is not a reply.

CPF2450
Work station message queue &1 not allocated to job.

CPF2451
Message queue &1 is allocated to another job.

CPF247A
Call stack entry not found.

CPF247E
CCSID &1 is not valid.

CPF2471
Length of field not valid.

CPF2477
Message queue &1 currently in use.

CPF2479
Call stack entry not found.

CPF2482
Message type &1 not valid.

CPF24A3
Value for call stack counter parameter not valid.

CPF24A8
Value for wait time not valid.

CPF24B3
Message type &1 not valid.

CPF2531
Message file &1 in &2 damaged for &3.

CPF2532
Job message queue is damaged. Job log ended.

CPF2548
Damage to message file &1 in &2.

CPF2551
Message key and message type combination not valid.

CPF36F7
Message queue QSYSOPR is allocated to another job.

CPF8127
&8 damage on message queue &4 in &9. VLIC log-&7.

CPF8176
Message queue for device description &4 damaged.

RCVNETF (Receive Network File) Command Description

RCVNETF Command syntax diagram

Purpose

The Receive Network File (RCVNETF) command receives a network file and copies the records into a physical database file or a save file.

If the original file is a save file, it must be received into a save file. Once the file has been received, it is removed from the queue of network files. Before a file can be received, the file specified by the TOFILE parameter must already exist.

When a source physical file is sent, the source sequence number and change date in positions 1 through 12 of the record are sent with the file. These are kept if the file is received into a source physical file, and are truncated if the file is received into a nonsource physical file. When a file that was originally a nonsource physical file is received into a source physical file, the source sequence numbers are created and placed in front of the records.

If the file is a physical file, the record length of the to-file must be at least as large as the record length of the original file. If the record length of the to-file is larger than that of the original file, the records are padded to the end with the default record value for the to-file.

Restrictions:

1. A user with security officer authority can receive the files sent to any user. Users with other than security officer authority can receive only files sent to them or to their group profile.
2. The user must have read authority to the library containing the to-file, and use and add authority to the to-file. The following additional authority may be required:
   - Object management authority, if a member is added to the file.
   - Object management authority and delete authority, if a save file or existing physical file member is cleared.

Required Parameter

FROMFILE
Specifies the name of the file that is received. This is the name of the file on the sending system.

Optional Parameters

TOFILE
Specifies the qualified name of the file that receives the copied records.

Note: If the network file sent is a database file, the to-file must be a physical database file with a record length at least as large as that of the original file. If the file sent is a save file, the to-file must be a save file. Overrides to this file are ignored.

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

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

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

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

***FROMFILE:** The network file is received into a file of the same name as the file sent.

file-name: Specify the name of the file into which the network file is received.

FROMMBR
Specifies the name of the file member that is received.

***ONLY:** Only one member is received for this file. This parameter value is valid only if there is one member for the specified network file, or if the NBR parameter is used to uniquely identify a single member.

member-name: Specify the name of the member that is received. A member name cannot be specified if the file is a save file.

TOMBR
Specifies the database file member that receives the data.

***FROMMBR:** The data is received into a member with the same name as the member specified in the FROMMBR parameter.

***FIRST:** The first member in the file receives the copied records.

member-name: Specify the name of the member that receives the records. A member name cannot be specified if the file is a save file.

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.

NBR Specifies the number of the file member that is received. This number is used to identify the member that is received when there is more than one member of the same name available for the file.

***LAST:** The last network file member with the specified member name is received. The last member is determined as the last member to arrive at the user's system; both FROMFILE and FROMMBR parameter values are used to determine the last network file member.

Note: The file member that arrived last at the user's system may not have been the last one sent by the sending user. The network does not guarantee the arrival sequence of separately sent files.

***ONLY:** Only one file member of the specified file name is received. If more than one member of that name is available, an escape message is sent, and the command is not run.

member-number: Specify the number of the member that is received.

USER Specifies the user to whom the file was sent.

***CURRENT:** The user profile that is currently running is used.

user-name: Specify the name of the user to whom the files were sent. Only a user with security officer authority can specify a name other than the user's own or the user's group profile.

FROMTYPE
Specifies the type of file that is received. This option should be used mainly when the file is an iSeries 400 computer, AS/400 system, or System/38 source file that was sent by a System/370 Virtual Machine (VM) or Multiple Virtual Storage (MVS) user. Since VM or MVS cannot identify whether the file is a source file, the user must specify that the file is a source file or a nonsource file.

***NETFILE:** The network file type is used to determine whether file type conversion is needed.

If the file is a nonsource file and is:

- Received into a nonsource file, the file is received unchanged.
- Received into a source file, the sequence numbers and date fields are added.

If the file is a source file and is:

- Received into a nonsource file, the sequence numbers and date fields are removed (the first 12 bytes of each record).
- Received into a source file, the file is received unchanged.

***SRC:** The file being received is a source file. The sequence numbers and date fields are in the file. If the file is received into another source file, the sequence numbers and date fields are not added to the file being received. If the file is received into a nonsource file, the sequence numbers and date fields are removed from the file.

Note: *SRC must *not* be specified if the network file does not contain sequence numbers and date fields in the first 12 bytes of each record.

Examples for RCVNETF

Example 1: Receiving a Member

```
RCVNETF   FROMFILE(FILEA)  TOFILE(FILEB/FILEA)
  FROMMBR(PAYROLL)
```

This command receives member PAYROLL of file FILEA into member PAYROLL of file FILEA in library FILEB. If there is an existing member of that name, the records in the member are replaced. If multiple members of that name are available, the last one to arrive at the destination system is received.

Example 2: Receiving a Network File

```
RCVNETF   FROMFILE(PERSONNEL)  NBR(*LAST)  USER(USR1)
```

This command receives a network file named PERSONNEL, which was sent to user USR1, into a file with the same name. Because the FROMMBR parameter is not specified, there must be only one member name available for this file. Because USR1 is specified, only someone with a user profile of USR1, someone with a group profile of USR1, or someone with security officer authority can use this command.

Example 3: Receiving a Source File

```
RCVNETF   FROMFILE(FILEA)  TOFILE(FILEB/FILEA)
  FROMMBR(PAYROLL)  FROMTYPE(*SRC)
```

This command specifies that the file being received is a source file and the sequence numbers and date fields are not added to the file being received.

Error messages for RCVNETF

***ESCAPE Messages**

CPF2204
User profile &1 not found.

CPF2207
Not authorized to use object &1 in library &3 type *&2.

CPF4101
File &2 in library &3 not found or inline data file missing.

CPF5715
File &1 in library &2 not found.

CPF8057
File &1 in &2 not a physical file or save file.

CPF8059
Member name not allowed for save file.

CPF8060
No files compare to the specified selection.

CPF8062
Record length of network file larger than receiving file.

CPF8063
Cannot assign necessary resource.

CPF8070
Not allowed to process files for user &1.

CPF8077
More than one file with same name found. See previously displayed messages.

CPF8080
MBROPT(*ADD) not allowed for save file.

CPF8081
File &5 member &6 number &7 already processed.

CPF8082
Cannot get network file &5 member &6 number &7.

CPF9005
System resource required to complete this request not available.

CPF9006
User not enrolled in system distribution directory.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

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

CPF9830
Cannot assign library &1.

CPF9843
Object &1 in library &3 type &2 cannot be accessed.

CPF9845
Error occurred while opening file &1.

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

CPF9847
Error occurred while closing file &1 in library &2.

CPF9848
Cannot open file &1 in library &2 member &3.

CPF9849
Error while processing file &1 in library &2 member &3.

RCVTIEF (Receive Technical Information Exchange File) Command Description

RCVTIEF Command syntax diagram

Purpose

The Receive Technical Information Exchange File (RCVTIEF) command receives files transmitted from the remote support network.

Required Parameters

LIB Specifies the library where the files are stored.

<u>*LIBL:</u> The library list is used to locate the file.

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

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

Optional Parameters

TYPE Specifies the types of files that are received.

***ALL:** All available files are received.

***OTHER:** Files with unspecified contents are received.

***SAVF:** Save files are received.

OUTPUT

Specifies whether the output from the command is shown at the requesting workstation or printed with the job's spooled output. More information on this parameter is in commonly used parameters.

***NONE:** A list of files received is not printed.

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

MAXRCDS

Specifies the maximum size (number of records) of any file that can be received.

10000: The maximum file size is 10000 records.

***NOMAX:** The system maximum is used.

number: Specify the maximum file size that can be received.

Example for RCVTIEF

```
RCVTIEF  LIB(MAIL)  TYPE(*OPEN)  OUTPUT(*PRINT)
  MAXRCDS(1000)
```

This command receives from TIE all OPEN files (any file except a save file). A list of the received files is printed. If any of the received files are larger than 1000 records, the RCVTIEF command fails. If all OPEN files are received successfully, they are removed from the mailbox.

No error messages.

RCLACTGRP (Reclaim Activation Group) Command Description

RCLACTGRP Command syntax diagram

Purpose

The Reclaim Activation Group (RCLACTGRP) command deletes a specified activation group and frees the resources that are scoped to it. It does not reclaim resources scoped to the job or scoped to the default activation group. This command is normally used only in the controlling program of the application.

An activation group is eligible to be reclaimed if it meets the following criteria:

- The activation group is not the default activation group.

  The default activation group cannot be reclaimed.
- The activation group is not active.

  An activation group cannot be reclaimed if there are programs or procedures running within the activation group.
- The activation group is not one of the debug activation groups.

  When the job is in debug mode, the activation groups in use do not appear as active on the Call Stack or Display Activation Group displays.
- The activation group is not a shared activation group.

A shared activation group cannot be reclaimed because it may be in use by another job.

When an activation group is reclaimed, all resources within the scope of the activation group are reclaimed. Resources within the scope of the activation group include static storage for programs in the activation group, open files, user interface manager (UIM) application resources, Common Programming Interface (CPI) Communications conversations, hierarchical file systems (HFS) resources, user-defined communications sessions, and pending changes for the commitment definition.

A close option can be specified on this command, and is used when closing mixed, communications, binary synchronous (BSC), and intersystem communications function (ICF) files. If an activation group level commitment definition has been started for the activation group, and it has pending committable changes, the close option also indicates whether the system implicitly commits or rolls back the pending changes before ending the commitment definition. When specifying a close option of *NORMAL, and there are no errors when closing files using the activation group level commitment definition, a commit is performed. Otherwise, a rollback is performed. See the Backup, Recovery, and Availability topic in the Information Center for information on how the system performs the rollback operation under commitment control.

An activation group should only be reclaimed if it will never be needed again within the same job. Otherwise, errors and unpredictable results may occur if other programs later attempt to access the resources that were reclaimed. Therefore, this command should normally only be used in the controlling program of an application. Specifying ACTGRP(*ELIGIBLE) requires full knowledge of the job environment. Otherwise, unpredictable results can occur.

For more information on the appropriate use of this command, see the ILE Concepts book.

Restrictions:

1. This command is not threadsafe. This command fails in a multithreaded job.

Required Parameter

ACTGRP

Specifies the activation group to be reclaimed.

***ELIGIBLE:** All eligible activation groups within the job are deleted.

activation-group-name: Specify the activation group to be reclaimed. The activation group can only be reclaimed if it has no active calls. If active calls exist, a message is displayed informing the user that the request failed. If the activation group is not found, a message is displayed informing the user that the request failed because the activation group was not found.

Optional Parameter

OPTION

Specifies whether to commit or roll back pending changes for an activation group level commitment definition, and whether a normal or abnormal close notification is sent to the attached host system when mixed, communications, BSC, and ICF files are closed. This parameter is ignored for all other files and objects within the scope of the activation group.

***NORMAL:** The changes pending for an activation group level commitment definition are committed (if there are no errors when closing files using the commitment definition), and a normal close notification is sent to the attached host system when mixed, communications, BSC, and ICF files are closed.

***ABNORMAL:** The changes pending for an activation group level commitment definition are rolled back and an abnormal close notification is sent to the attached host system when mixed, communications, BSC, and ICF files are closed.

Example for RCLACTGRP

```
RCLACTGRP  ACTGRP(MYGROUP)
```

This command reclaims the activation group MYGROUP.

Error messages for RCLACTGRP

***ESCAPE Messages**

CPF1653
Activation group &1 not found.

CPF1654
Activation group &1 cannot be deleted.

CPF1892
Function &1 not allowed.

RCLDDMCNV (Reclaim Distributed Data Management Conversations) Command Description

RCLDDMCNV Command syntax diagram

Purpose

The Reclaim Distributed Data Management Conversations (RCLDDMCNV) command reclaims all distributed data management (DDM) source system conversations that are not currently being used by a source job, even if the DDMCNV attribute value for the job is *KEEP. The command allows the user to reclaim unused DDM conversations without closing all open files or doing any of the other functions performed by the RCLRSC command. The RCLDDMCNV command applies only to the DDM conversations for the job on the *source* system in which the command is entered.

Although this command applies to *all* DDM conversations used by a job, using it does *not* mean that all of them are taken down. A conversation is taken down *only* if there are no active users for that conversation.

There are no parameters for this command.

Example for RCLDDMCNV

```
RCLDDMCNV
```

This command checks all DDM conversations for the job in which the command is entered, determines if there are any users of each conversation, and reclaims each one not being used.

Error messages for RCLDDMCNV

None

RCLDLO (Reclaim Document Library Object) Command Description

RCLDLO Command syntax diagram

Purpose

The Reclaim Document Library Object (RCLDLO) command is used to reclaim:

- A document.
- A folder.

- A folder and all documents and folders directly or indirectly contained within it.
- Internal document library system objects.
- Internal document library system objects, unfiled distribution documents, and all filed folders and documents on the system. Related document details are synchronized.
- Internal document library system objects and all filed folders and documents in one auxiliary storage pool (ASP). Related document details are synchronized.

Restrictions:

1. The user does not need authority to a document or folder to reclaim it.
2. Exclusive use of the document or folder is required while it is being reclaimed.
3. The user does not need to be enrolled in the system distribution directory.
4. Reclaiming internal document library system objects, all document library objects, or all document library objects in one ASP can be done only when no other jobs are using folders or documents. The user of this command may receive an error message indicating that internal objects are locked. This means that another user is using document library functions which cannot run at the same time as the RCLDLO command. Try this command again after other document library activity has ended.

Note: The iSeries 400 PC/Support jobs and several jobs which run in the QSNADS subsystem work with documents and folders. These jobs can prevent RCLDLO options that reclaim internal system objects because they obtain locks on internal system objects. All jobs running in the QSNADS subsystem and iSeries 400 PC/Support jobs should be ended before attempting to reclaim internal system objects.

Required Parameter

DLO Specifies the document library object to reclaim.

***FLR:** A folder, and all folders and documents directly or indirectly within it, are to be reclaimed.

***SYSOBJNAM:** A system object name is used to identify the folder or document to reclaim. DLO(*SYSOBJNAM) must be specified when reclaiming a folderless document, an internal document, or a distribution document. It can also be specified instead of a folder or document name if the system object name is known.

***INT:** Internal document library system objects are to be reclaimed.

Note: The internal document library system objects are used to manage the documents and folders on the iSeries 400. RCLDLO DLO(*INT) is only necessary if the internal objects become damaged. If the internal objects are damaged, attempts to access documents and folders will result in the message CPF8A46 (Internal system objects are damaged), possibly followed by the message CPF9032 (Document interchange session not started),

***DOCDTL:** Internal document library system objects and document details are to be reclaimed. DLO(*DOCDTL) synchronizes the relationships between all document library objects and their document details and will fix inconsistencies between them.

Note: The RCLDLO DLO(*DOCDTL) command can be a long-running function, performing a subset of the RCLDLO DLO(*ALL) processing necessary to guarantee consistency between internal system objects, document details, and DLOs.

***ALL:** All documents and folders (as specified on the ASP parameter) are reclaimed in addition to DLO(*DOCDTL) processing.

Note: The RCLDLO DLO(*ALL) command can be a long-running function, depending on the number of documents and folders on the system. If the RCLDLO command can be issued at the user's discretion, the user may wish to avoid the operation until the time required can be scheduled.

document-library-object-name: Specify the name of the folder or document to reclaim.

Optional Parameters

FLR Specifies the name of the folder that contains the document.

Note: A folder name is entered in this parameter only if *FLR, or a folder or document name is entered in the DLO parameter.

***NONE:** The folder or document is not within a folder.

folder-name: Specify the name of the folder that contains the document or folder to reclaim, or the name of the folder to reclaim along with all documents and folders within it.

SYSOBJNAM
Specifies the system object name. This parameter is valid only when DLO(*SYSOBJNAM) or DOCL(*SYSOBJNAM) is specified. A full ten characters must be specified.

SYSOBJATR
Specifies the attributes of the object to reclaim. A value other than *NONE can be entered in this parameter only if *SYSOBJNAM is entered in the DLO parameter.

***NONE:** The object to reclaim is a filed document or folder.

***INTDOC:** The object to reclaim is an internal document.

***DST:** The object to reclaim is a distribution document.

ASP Specifies the identifier (ID) of the auxiliary storage pool (ASP) of the document library object to be reclaimed. A value other than *ANY can be specified on this parameter only if *ALL or *DOCDTL is specified on the DLO parameter.

***ANY:** The objects to be reclaimed reside in any ASP. When *ALL is specified on the DLO parameter, all document library objects on the system are reclaimed.

ASP-ID: Only document library objects that reside in the specified ASP are to be reclaimed. All document library objects in other ASPs are ignored. Valid values range from 1 through 16 and must designate an existing ASP that contains document library objects. ASP 1 is the system ASP.

Note: Unfiled distribution documents are classified as document library objects in the system ASP.

OUTFILE

Specifies the name of the database file to which special output is directed. If the output file does not exist, this command creates a database file in the specified library. If the file is created by this function, the descriptive text is “OUTFILE created by RCLDLO” and the authority for users without specific authority to the file is *EXCLUDE. A value other than *NONE can be specified on this parameter only if *ALL or *DOCDTL is specified on the DLO parameter.

The output directed to this file includes the names of any documents that are physically damaged (and therefore unusable) or documents or folders that were missing from the system (and for which the document details have been removed). This file is intended to provide the user with a record of what was lost (such as a user ASP) when recovering from hardware failure.

***NONE:** No output is directed to a database file.

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

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

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

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

database-file-name: Specify the qualified name of the database file that is to receive the output. This file can be reused when other RCLDLO commands are issued. Output is added to the file as specified on the OUTMBR parameter. The IBM-supplied database file, QSYS/QARCLDLO, cannot be specified.

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

Examples for RCLDLO

Example 1: Reclaiming a Folder

```
RCLDLO   DLO(FLR1)
```

This command reclaims folder FLR1.

Example 2: Reclaiming a Document Within a Folder

```
RCLDLO   DLO(A)  FLR(FLR2)
```

This command reclaims folder or document A in folder FLR2.

Example 3: Reclaiming a Folder and All Documents and Folders Within It

```
RCLDLO   DLO(*FLR)  FLR(FLR3)
```

This command reclaims folder FLR3 and all folders and documents directly or indirectly contained within it.

Example 4: Reclaiming an Internal Document

```
RCLDLO   DLO(*SYSOBJNAM)  SYSOBJNAM(AMBT133080)
  SYSOBJATR(*INTDOC)
```

This command reclaims the internal document specified by the system object name AMBT133080.

Example 5: Reclaiming a Distribution Document

```
RCLDLO   DLO(*SYSOBJNAM)  SYSOBJNAM(AMBT133082)
  SYSOBJATR(*DST)
```

This command reclaims the distribution document specified by the system object name AMBT133082.

Example 6: Reclaiming Document Library System Objects

```
RCLDLO   DLO(*INT)
```

This command reclaims internal document library system objects.

Example 7: Reclaiming Document Library System Objects and Document Details

```
RCLDLO   DLO(*DOCDTL)
```

This command reclaims internal document library system objects and document details for all folders and documents.

Example 8: Reclaiming Document Library System Objects and All Documents and Folders

```
RCLDLO   DLO(*ALL)
```

This command reclaims internal document library system objects and all documents and folders and synchronizes their document details.

Example 9: Reclaiming Document Library System Objects and All Documents and Folders in an ASP

```
RCLDLO   DLO(*ALL)  ASP(2)
```

This command reclaims internal document library system objects and all documents and folders residing in ASP 2 and synchronizes their document details.

Additional Considerations

When RCLDLO DLO(*ALL) or DLO(*DOCDTL) is run, the following items indicate what happens for folders and documents whose document details are updated.

- Objects that have lost folder path information are associated with a system reclaim folder. An informational message is sent to QSYSOPR message queue for each object reclaimed in this manner. System reclaim folders, recognized by the name ’QRCLnnnn.DOC’ or ’QRCLnnnn.FLR,’ should be deleted from the system after moving the objects within them to an appropriate user folder. It may also be necessary to use the Edit Document Library Object Authority (EDTDLOAUT) command to grant specific authority to the reclaimed objects again.

Note: System reclaim folders are created in the ASP of the object or objects they are to contain.

- A log of the actions taken by the RCLDLO DLO(*ALL) or DLO(*DOCDTL) command is sent to the joblog and a listing of status messages is sent to the QHST system log. Additional information about the objects shown in the log can be displayed by the Display Object Description (DSPOBJD) command.
- If the OUTFILE option is used on RCLDLO DLO(*ALL) or DLO(*DOCDTL), records for every document or folder which was physically damaged (permanently unusable) or destroyed are placed into the specified file. This provides the system administrator with a list that can be used to determine which documents and folders need to be restored from backup media.

Error messages for RCLDLO

***ESCAPE Messages**

CPF8AA4
Reclaim of &2 not successful.

CPF8A29
Reclaim of &2 partially successful.

RCLLIB (Reclaim Library) Command Description

RCLLIB Command syntax diagram

Purpose

The Reclaim Library (RCLLIB) command rebuilds the internal objects of a library that contain the object descriptive information for all objects in the library and the library object itself. The library size is not reduced as a result of running this command. If the library does not contain any objects and you want to reduce the size of the library, you must delete and recreated the library.

This command rebuilds, where possible, internal objects of the library that were damaged or destroyed.

Restrictions:

1. The user must have *OBJEXIST and *USE authority to the library being rebuilt. This is the same authority required to delete a library with the DLTLIB command.
2. Only the internal objects of a library which contain the object descriptive information are rebuilt. No other objects in the library are validated or rebuilt.
3. This command can be used on any library other than QTEMP.

Parameters

LIB Specifies the name of the library being rebuilt.

Example for RCLLIB

```
RCLLIB  LIB(TESTLIB)
```

This command determines if the object descriptive information of library TESTLIB is damaged. The damaged parts of the library are rebuilt.

RCLOPT (Reclaim Optical) Command Description

RCLOPT Command syntax diagram

Purpose

The Reclaim Optical (RCLOPT) command rebuilds the optical index database or the internal library indexes after they have been damaged or destroyed. These database files include the optical volume index (QAMOVAR) and the optical directory index (QAMOPVR) in library QUSRSYS. This command requires exclusive use of the directly attached optical device.

Restriction: You must have *USE authority to use this command. It is shipped with *EXCLUDE public authority.

Required Parameter

MLB Specifies the name of the directly attached optical device for which the optical indexes are rebuilt.

***ALL:** The optical indexes for all directly attached optical devices are rebuilt.

optical-media-library: Specify the name of the directly attached optical device for which the optical indexes are rebuilt.

Optional Parameters

OPTION
Specifies which type of reclaim operation is performed.

***SYNC:** The optical volume index is synchronized with the internal device index. If a volume is not in the optical volume index, both the volume and the directory index entries are created.

***UPDATE:** The optical volume index and the optical directory index are rebuilt with information read from the optical cartridge. The optical directory index parameter default is set to *YES.

***RESET:** The internal device index is rebuilt with information read from the optical cartridge. The optical volume index and the optical directory index are then rebuilt from the internal device index.

VOL Specifies which volumes are used during the reclaim operation when OPTION(*UPDATE) is specified.

***ALL:** All volumes in the optical device are used.

volume-identifier: Specify the volume identifier of a specific volume to use during the reclaim operation.

DIR Specifies whether the optical directory index (QAMOPVR) is rebuilt when OPTION(*UPDATE) is specified.

***YES:** The optical directory index is rebuilt.

***NO:** The optical directory index is not rebuilt.

Example for RCLOPT

```
RCLOPT   MLB(OPT01)  OPTION(*UPDATE)  VOL(VOL01)
```

This command re-creates both the optical volume index and the optical directory index for the optical volume VOL01 in the optical media library OPT01.

Error messages for RCLOPT

***ESCAPE Messages**

OPT0125
Command &1 completed with errors, more information in job log.

OPT1320
Optical volume &1 in use.

OPT1325
Optical volume format not recognized.

OPT1330
Optical volume not found or not useable.

OPT1331
Optical volume &1 not found.

OPT1346
Operation not allowed to volume located in a remote optical device.

OPT1360
Media directory corrupted on optical volume &1.

OPT1460
Optical volume &1 is not in an optical device.

OPT1530
&1 does not represent a valid optical device.

OPT1555
Optical device &1 in use.

OPT1605
Media or device error occurred.

OPT1640
Error occurred reading files or directories.

OPT1652
Device &1 is not an optical media library.

OPT1790
Operation not allowed or conflicts with another request.

OPT1805
Error accessing optical volume index file.

OPT1810
Error accessing optical directory index file.

OPT1815
Internal program error occurred.

OPT1820
Internal error occurred on optical device &1.

OPT1821
Internal error occurred on optical device &1.

OPT1825
Optical indexes are incorrect for optical device &1.

OPT1860
Request to optical device &1 failed.

OPT1861
No device description configured for resource &1.

OPT1862
No active device description for resource &1.

OPT1863
Optical libraries need to be reclaimed.

OPT1872
Optical request timed out.

OPT2040
Error accessing backup control file.

OPT2155
Reclaim Optical failed for optical device &1.

OPT2165
Reclaim Optical failed for volume &1.

OPT2187
Optical index inconsistency, reclaim optical required.

OPT2188
Optical index files are damaged.

OPT2190
Error occurred during reclaim optical processing while accessing volume &1.

OPT2191
Error occurred while reading volume directory for volume &1.

OPT2301
Internal system object in use.

OPT7740
User not authorized to object &2 in library &3 type &4.

RCLRSC (Reclaim Resources) Command Description

RCLRSC Command syntax diagram

Purpose

The Reclaim Resources (RCLRSC) command is intended for use by the controlling program in an application to free resources that were allocated by programs that have finished running and returned control to the controlling program. The resources used by these programs can then be used by other programs running on the system. Based on the reference level specified by the LVL parameter, this command reclaims resources that are scoped to the default activation group. This command does not reclaim resources that are scoped to the job or resources that are scoped to any activation group other than the default activation group. The resources that are reclaimed by this command are

- static storage

  For an original program model (OPM) program, its static storage will be reclaimed. For a user state Integrated Language Environment (ILE) program, its static storage will be marked so that it will be re-initialized on the next call. Static storage is not affected for a service program, a system state ILE program, or any ILE program running in an activation group other than the default activation group.
- open files of object type *FILE
- user interface manager (UIM) application resources
- Common Programming Interface (CPI) Communications conversations
- hierarchical file systems (HFS) resources
- user-defined communications sessions.

In addition, after distributed data management (DDM) files are closed based on the criteria described above, all DDM source system conversations in this job that are not currently in use will be ended, regardless of the activation group or reference level from which they were started.

The RCLRSC command is *not* needed to reclaim the files and static storage of most programs, such as CL programs that end (return) normally, RPG programs that have the last record (LR) indicator set on, and COBOL programs. The RCLRSC command should *not* be used if it might be processed while any COBOL program is still active in the application.

For more information on the appropriate use of this command, see the documentation associated with the programming language(s) used in the application.

Restrictions:

1. This command is not threadsafe. This command fails in a job that allows multiple threads.
2. Do not specify LVL(*CALLER) on this command if it is used in a CL program that also uses the Send File (SNDF), Receive File (RCVF), or Send Receive File (SNDRCVF) commands. Specifying RCLRSC LVL(*CALLER) in such a program causes unpredictable results when the SNDF, RCVF, or SNDRCVF commands are used after the program runs.
3. Do not specify LVL(*CALLER) on this command if it is issued from a command line, as this will result in unpredictable results.

Optional Parameters

LVL Specifies the reference level at which resources are reclaimed.

*: The reference level is the program or procedure that contains this RCLRSC command. The resources are reclaimed for programs or procedures that have finished running and returned control to this program.

*CALLER: The reference level is the program or procedure that called the program or procedure containing this RCLRSC command. This value allows controlling programs or procedures written in a high-level language to call a CL program to reclaim resources to the level of the controlling program or procedure. The effect is the same as if the command were issued from the controlling program or procedure.

Note: Using the *CALLER value can cause unexpected results when running the RCLRSC command from a command line or from within a program that works with open files.

OPTION

Specifies whether a normal or abnormal close notification is sent to the attached host system when mixed, communications, binary synchronous (BSC), and intersystem communications function (ICF) files are closed. This parameter is ignored for all other files and objects.

*NORMAL: The attached host system is given a normal close notification when mixed, communications, BSC, and ICF files are closed.

*ABNORMAL: The attached host system is given an abnormal close notification when mixed, communications, BSC, and ICF files are closed. Use this when the controlling program detects error conditions that should be communicated to the host systems (the error condition need not be file-related).

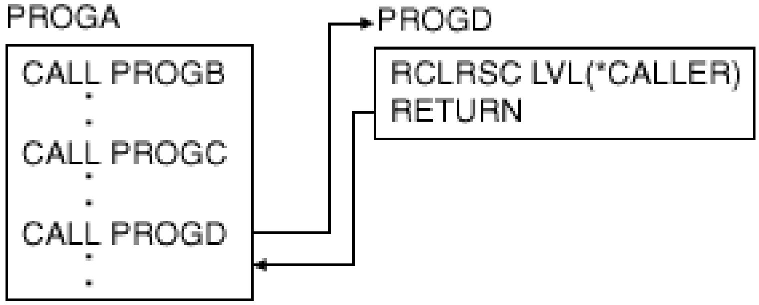
Examples for RCLRSC

Example 1 (OPM)

```
PROGA
    *
    *
```

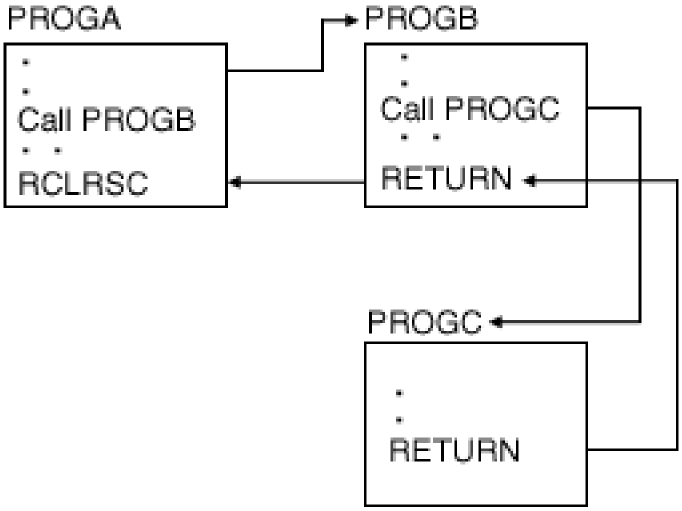
```
CALL PROGB
RCLRSC
     *
     *
CALL PROGC
RCLRSC
     *
     *
     *
```

In this example, PROGA is a controlling program in an application. PROGA calls other programs, which return control to PROGA when they have finished running. Because control is returned to the next sequential instruction, the RCLRSC command is issued following each CALL command to free the static storage that was used by the called program, and to close the files that were left open.



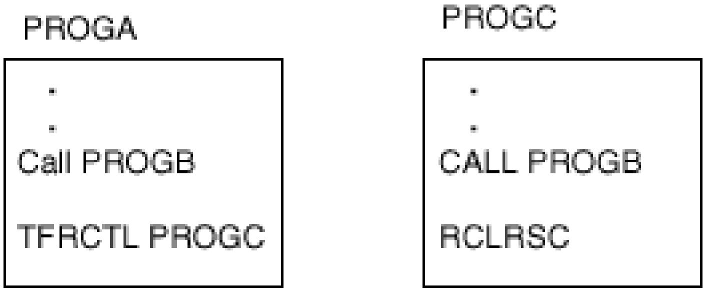
Example 2 (OPM)

In this example, PROGA is a controlling program that is written in another high-level language. The RCLRSC command cannot be issued from the high-level language program so PROGD, a CL program, is called to issue the command. When the RCLRSC command is issued in PROGD, the static storage used by PROGB and PROGC is freed; files that were left open are closed.



Example 3 (OPM)

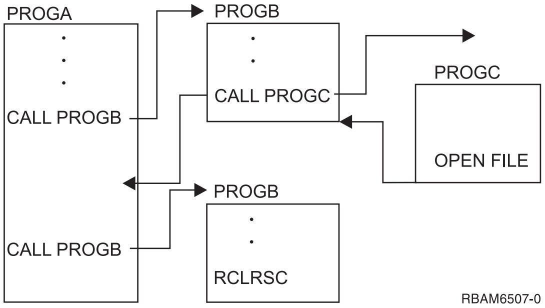
In this example, PROGA is a controlling program. When the RCLRSC command is issued, the static storage used by PROGB and PROGC is freed; files that were left open are closed.



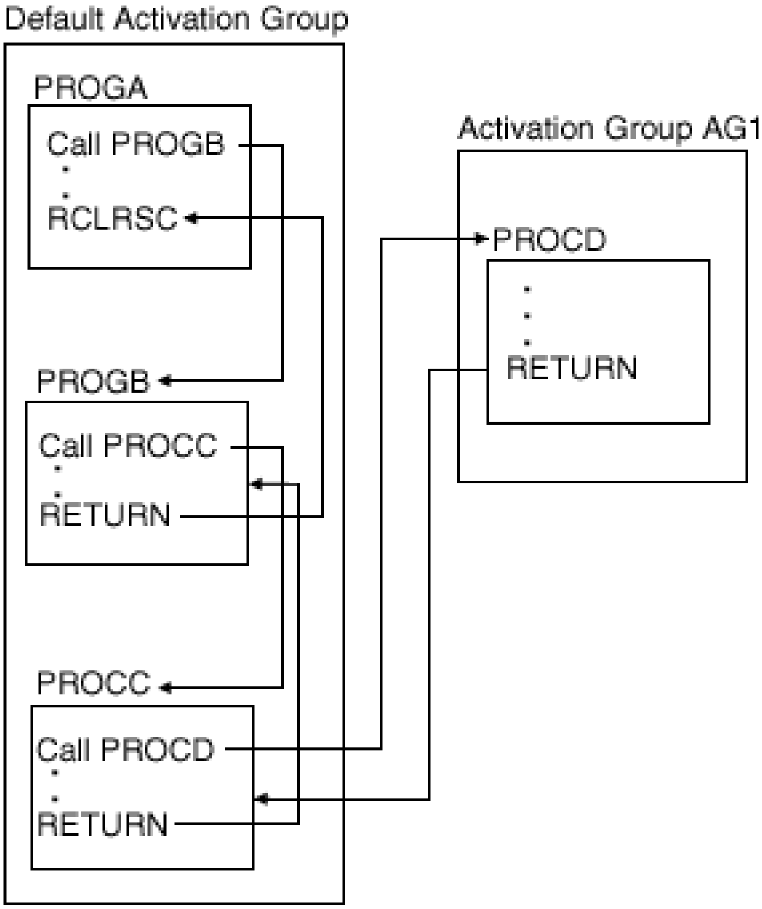
Example 4 (OPM)

In this example, PROGA calls PROGB and, after returning from PROGB, PROGA transfers to program PROGC. Because PROGB has already been called, static storage exists, and the call to PROGB from PROGC does not cause any new allocation for static storage; PROGC cannot reclaim the static storage used by PROGB. If PROGB opened files when it was called by PROGA, these files would remain open; if PROGB opened files when it was called by PROGC, these files are closed.

Example 5 (OPM)



In this example, PROGA calls PROGB, which in turn calls PROGC. PROGC opens a file. Control returns to PROGA. PROGA calls PROGB a second time, and PROGB invokes RCLRSC. Since RCLRSC is based upon the position in the stack of the current invocation of a program, the file opened by PROGC is not affected. PROGC was invoked earlier than the current PROGB. In order for RCLRSC to close the file, the RCLRSC command will have to be invoked by PROGA.



Example 6 (OPM & ILE)

This example shows how ILE procedures and activation groups are affected by the RCLRSC command.

In this example, PROGA is a program running in the default activation group. PROGA calls program PROGB which runs in the default activation group. PROGB calls ILE procedure PROCC which runs in the default activation group. PROCC calls ILE procedure PROCD which causes activation group AG1 to be created. PROCD returns to PROCC. PROCC returns to PROGB. PROGB returns to PROGA, which then calls the RCLRSC command.

PROGA calls the RCLRSC command. Any resources in use by PROGA are still open, since PROGA is still in use. Any resources by program PROGB or procedure PROCC are reclaimed, since the program and procedure ran in the default activation group and are no longer active. Any resources opened by procedure PROCD are left alone, since procedure PROCD ran in activation group AG1 and only the default activation group is affected by the RCLRSC command.

Any other use of the RCLRSC command can result in files remaining open and storage remaining allocated.

Error messages for RCLRSC

***ESCAPE Messages**

CPF1892
Function &1 not allowed.

RCLSPLSTG (Reclaim Spool Storage) Command Description

RCLSPLSTG Command syntax diagram

Purpose

The Reclaim Spool Storage (RCLSPLSTG) command reclaims unused storage for spooled files that have not been used for more than the number of days specified by the user. Spooled files are stored with database file members on the system. When a spooled file is deleted, the member is emptied but not deleted. Therefore, the member can be reused for the next spool file created. Reusing empty members improves the performance time when creating new spooled files. The RCLSPLSTG command deletes unused and empty database members. This command uses synchronous processing. More information about synchronous processing is in the Backup, Recovery, and Availability topic in the Information Center.

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

Required Parameters

DAYS Specifies that storage for spooled files that have not been used for the number of days specified by the user is reclaimed.

***NONE:** All unused storage for spooled files is reclaimed. If there is no unused storage when subsequent spooled files are created, then the system allocates storage for those spooled files. The allocation of storage for these spooled files may slow down the jobs that are creating them. More information is in the File Management topic in the Information Center.

days-to-keep-unused-storage: Specify the number of days to keep unused storage for spooled files. Storage that remains unused for more than the number of specified days is reclaimed.

Example for RCLSPLSTG

```
RCLSPLSTG  DAYS(30)
```

This command reclaims all unused storage for spooled files that have remained unused for more than 30 days. When storage has been unused for 1 second over 30 days it is reclaimed because a date and time stamp is placed on the storage area.

Error messages for RCLSPLSTG

None

RCLSTG (Reclaim Storage) Command Description

RCLSTG Command syntax diagram

Purpose

The Reclaim Storage (RCLSTG) command corrects, where possible, objects that were incompletely updated (such as database files, libraries, device descriptions, directories, and stream files) and user profiles containing incorrectly recorded object ownership information. Any unusable objects or fragments are deleted.

This command reclaims all objects secured by an authorization list that is damaged or destroyed and assigns the objects to the authorization list QRCLAUTL.

Because the amount of time required to run this command varies with the number of objects in auxiliary storage, the system periodically sends messages to the work station where the command was specified.

The RCLSTG command can also be used to reclaim storage when, during an IPL, not enough storage is available to make the system fully operational. In that case, the system operator can specify the command immediately after receiving the message about insufficient storage.

If very little additional auxiliary storage is available, the system overhead required to run the RCLSTG command may need more than the remaining storage; in that case, the RCLSTG command fails.

Note: The RCLSTG command can be a long-running function, depending on the number and type of objects in the system, and the amount of damage to them. Because RCLSTG touches each object multiple times, having enough memory can significantly reduce the time required to run RCLSTG. Conversely, having too little memory can lead to storage thrashing which can significantly increase the time required to run RCLSTG. If database file objects are damaged, the keyed access paths may need to be rebuilt; that operation takes a substantial amount of time. If the RCLSTG command can be run at the user's discretion, the user may want to avoid the operation until the required time can be scheduled.

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.
2. If option *SYSBAS is specified on the ASPDEV parameter, all subsystems must be inactive before the RCLSTG command can be specified. The End System (ENDSYS) or End Subsystem (ENDSBS) command with *ALL specified on the SBS parameter can be used to make the subsystems inactive. You must have job control (*JOBCTL) authority to use the ENDSBS or ENDSYS command.
3. Only permanent objects in auxiliary storage are reclaimed; temporary objects are reclaimed by running a system initial program load (IPL).
4. Before running the RCLSTG command after an IPL, you may need to wait several minutes for the IPL to complete. Use the Work with Active Jobs (WRKACTJOB) command to verify that no jobs are running.
5. If option *SYSBAS is specified on the ASPDEV parameter, this job must be in the controlling subsystem and must be the only job active in the system.
6. » If an auxiliary storage pool (ASP) device or an ASP group is specified on the ASPDEV parameter, it cannot have active users or be in use by a different job. «

Optional Parameters

SELECT
Specifies the reclaim functions that are performed.

***ALL:** All reclaim functions are performed.

***DBXREF:** The database cross-reference table reclaim function is performed.

OMIT Specifies the reclaim functions to be omitted from the reclaim operation.

***NONE:** No reclaim functions are omitted.

***DBXREF:** The database cross-reference table reclaim function is omitted.

ASPDEV
Specifies the auxiliary storage pool (ASP) to be reclaimed.

***SYSBAS:** The system ASP and all basic ASPs are reclaimed. The system ASP has an ASP number of 1. Basic ASPs have ASP numbers of 2 through 32.

auxiliary-storage-pool-device-name: The specified ASP device is reclaimed. ASP devices have ASP numbers greater than 32. Reclaim storage for an ASP device can be run without being in restricted state. You can submit multiple jobs, each performing RCLSTG on a different ASP device, to reclaim multiple ASP devices in parallel.

» *auxiliary-storage-pool-group-name:* The primary ASP and all the secondary ASPs within the ASP group are reclaimed. The name of any ASP within the ASP group is accepted. Primary and secondary ASPs have ASP numbers greater than 32. Reclaim storage for an ASP group can be run without being in restricted state. You can submit multiple jobs, each performing RCLSTG on a different ASP group, to reclaim multiple ASP groups in parallel. «

Examples for RCLSTG

Example 1: Reclaim Storage of the System ASP and All Basic ASPs

```
RCLSTG
```

This command, specified interactively, locates all system objects created before the last initial program load (IPL). Objects without owners are given default owners, and those that are lost from their specified libraries are inserted into the QRCL library or the default library, or are deleted.

Objects that are lost from their specified directories are inserted into the ’/QReclaim’ directory (if the object was originally located in the root file system) or the ’/QOpenSys/QReclaim’ directory (if the object was originally located in the QOpenSys file system).

Lost objects that are deleted are certain user objects and certain OS/400 system objects that are damaged and not usable.

The QRCL library, which is created (when needed) by the RCLSTG command, is a permanent library.

The ’/QReclaim’ and ’/QOpenSys/QReclaim’ directories, which are created (when needed) by the RCLSTG command, are permanent directories; but if they contain no objects at the end of the operation because they were all reclaimed, the directories are deleted.

Example 2: Reclaim Storage to Reclaim the Database Cross-reference Table

```
RCLSTG   SELECT(*DBXREF)
```

This command reclaims the database cross-reference table.

Example 3: Reclaim Storage of the Entire System That Omits the Reclaim of the Database Cross-reference Table

```
RCLSTG   OMIT(*DBXREF)
```

This command performs all reclaim storage functions but omits the reclaim of the database cross-reference table.

Example 4: Reclaim Storage of an ASP device

```
RCLSTG   ASPDEV(MYASPDEV)
```

This command reclaims storage for the ASP device name MYASPDEV.

Error messages for RCLSTG

*ESCAPE Messages

CPF180B
Function &1 not allowed.

CPF180C
Function &1 not allowed.

CPF2119
Library &1 locked.

CPF2120
Cannot delete library &1.

CPF2126
Attempt to recover library &1 failed.

CPF2127
User profile &2 damaged.

CPF5729
Not able to allocate object &1.

CPF8201
User profile &1 does not exist or is damaged.

CPF8204
Commitment control cannot be active during reclaim storage.

CPF8205
Library &1 does not exist or is damaged.

CPF8209
System not in proper state to reclaim storage.

CPF8210
Reclaim storage failed for auxiliary storage pool device &2.

CPF8211
Library &1 damaged. RCLSTG command ended.

» **CPF8212**
SELECT(*DBXREF) or OMIT(*DBXREF) invalid when reclaiming an User Defined File System (UDFS) ASP.«

CPF8224
Duplicate object found while moving or renaming member.

CPF8251
RCLSTG command ended. Library &1 damaged.

CPF8252
Error occurred during rebuild of damaged library &1.

» **CPF9814**
Device &1 not found.

CPFB8ED
Device description &1 not correct for operation.«

*STATUS Messages

CPI8206
&1% of objects » read from disk. «

CPI8210
Processing data base relationships.

CPI8212
Data base/library/directory recovery in progress.

CPI8213
Processing objects on the system.

CPI8214
All permanent objects have valid owners.

CPI8215
Object description verification in progress.

CPI8216
Final clean up in progress.

CPI8217
Mail Server Framework cleanup in progress.

CPI8218
Directory recovery in progress.

CPI8219
Directory cleanup in progress.

CPI8220
Message queue QSYSOPR in *HOLD delivery mode.

» **CPIA916**
&1% of directories processed.«

RCLTMPSTG (Reclaim Temporary Storage) Command Description

RCLTMPSTG Command syntax diagram

Purpose

The Reclaim Temporary Storage (RCLTMPSTG) command removes the temporarily decompressed copies of panel groups, menus, display files, and printer files, thereby freeing up system storage space.

- *Compressed Objects* are objects that consume less storage space than decompressed objects. When a compressed object is used or a compressed program is called, a decompressed version of the object automatically becomes available to the user.
- *Decompressed Objects* are objects that use the system storage space allocated to them and are in a final, ready-to-use state.
- *Temporarily Decompressed Objects* are temporarily decompressed copies of compressed objects. The system allocates storage space for the decompressed objects, which is consumed by the temporary copies until the system or the user determines that the temporary storage space needs to be reclaimed.

  Temporary storage is automatically reclaimed when:

  – The RCLTMPSTG command is run

  – The next initial program load (IPL) is run

  – The object is used often enough to cause the system to permanently decompress it

  When an object is permanently decompressed, the compressed version of the object is destroyed as well as any temporary forms of the object; however, compressed versions remain intact as long as the objects are temporarily decompressed.

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.
2. The user must have object management authority to the object specified and execute authority to the library.

Optional Parameters

LIB Specifies the name of the library in which allocated storage space for temporarily decompressed objects is reclaimed.

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

***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.

» ***ALLUSR**: User libraries are all libraries with names that do not begin with the letter Q except for the following:«

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

» 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
QGPL
QGPL38
QMPGDATA
QMQMDATA
QMQMPROC
QPFRDATA
QRCL
» QRCLxxxxx«
» QSYS2«
» QSYS2xxxxx«
QS36F
QUSER38
QUSRADSM
QUSRBRM
QUSRDIRCL
QUSRDIRDB
QUSRIJS
QUSRINFSKR
QUSRNOTES
QUSROND
QUSRPOSGS
QUSRPOSSA
QUSRPYMSVR
QUSRRDARS
QUSRSYS
QUSRVI
QUSRVxRxMx

Notes:

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

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

DAYS Specifies the number of days an object has not been used or changed. If a temporarily decompressed object has not been used or changed for more than the specified number of days, it is reclaimed. If it has been used or changed, it is left temporarily decompressed.

7: Objects that have not been used or changed for more than seven days are reclaimed.

***NONE:** The storage space allocated for a temporarily decompressed object is reclaimed regardless of the number of days it has not been used or changed, unless the object is in use when this command is run.

number-of-days: Specify the number of days a temporarily decompressed object can be unused before the temporary space allocated for it is reclaimed. Valid values range from 1 through 366.

Example for RCLTMPSTG

```
RCLTMPSTG   LIB(QGPL)
```

This command reclaims the space consumed by all of the temporarily decompressed copies of objects in library QGPL that have not been used or changed in the last 7 days.

Error messages for RCLTMPSTG

***ESCAPE Messages**

CPF2110
Library &1 not found.

CPF2113
Cannot allocate library &1.

CPF2176
Library &1 damaged.

CPF3B07
&1KB storage reclaimed, &5 objects not processed.

CPF9838
User profile storage limit exceeded.

RLSCMNDEV (Release Communications Device) Command Description

RLSCMNDEV Command syntax diagram

Purpose

The Release Communications Device (RLSCMNDEV) command restores the communications capability of a specified device held by the Hold Communications Device (HLDCMNDEV) command.

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

Required Parameters

DEV Specifies the name of the device whose communications are released after being held. Specify the name of the device. Devices whose communications can be held by the HLDCMNDEV command are:

DEV Value	Device
3180	Display station
3277	Display station
3278	Display station
3279	Display station
3287	Printer (work station)
5219	Printer (work station)
5224	Printer (work station)
5225	Printer (work station)
5251	Display station
5252	Display station

5256	Printer (work station)
5291	Display station
5292	Display station
PLU1	Primary logical unit, type 1 (for SNA)
BSC	Binary synchronous device (Base and RJE)
BSCT	This iSeries 400 is a BSC multipoint tributary station
APPC	Logical unit in advanced program-to-program communications network

Example for RLSCMNDEV

```
RLSCMNDEV  DEV(WSPR05)
```

This command restores the communications capability of the currently held device WSPR05.

Error messages for RLSCMNDEV

***ESCAPE Messages**

CPF5920
Device &1 varied off or in diagnostic mode.

CPF5921
Device &1 not a communications device.

CPF5935
Error occurred during command processing.

CPF5984
Not authorized to perform function.

CPF9814
Device &1 not found.

CPF9825
Not authorized to device &1.

RLSDSTQ (Release Distribution Queue) Command Description

RLSDSTQ Command syntax diagram

Purpose

The Release Distribution Queue (RLSDSTQ) command releases a distribution queue from a held status and allows it to be sent.

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

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
2. Messages that report errors about distribution queues may display or print different characters than the user 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 on the Work with Distribution Queue (WRKDSTQ) command. An

error may be reported if the character-string value specified for the 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).

Required Parameters

DSTQ Specifies the name of the distribution queue being released from being held. Both normal and high priority portions of the specified distribution queue are shown or printed. The queue specified must have been previously configured. See the Configure Distribution Services (CFGDSTSRV) command or the Add Distribution Queue (ADDDSTQ) command.

PTY Specifies whether the normal priority or high priority portion of the specified queue is released from being held.

***NORMAL:** Releases the normal priority queue, which is for distributions with a service level of data low.

***HIGH:** Releases the high priority queue, which is for distributions with a service level of fast, status, or data high.

Examples for RLSDSTQ

Example 1: Releasing the Normal Priority Portion of the Queue

```
RLSDSTQ  DSTQ(CHICAGO)  PTY(*NORMAL)
```

This command releases the normal priority portion of the CHICAGO distribution queue.

Example 2: Releasing the High Priority Portion of the Queue

```
RLSDSTQ  DSTQ(ATLANTA)  PTY(*HIGH)
```

This command releases the high priority portion of the ATLANTA distribution queue.

Error messages for RLSDSTQ

*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.

RLSIFSLCK (Release Integrated File System Locks) Command Description

RLSIFSLCK Command syntax diagram

Purpose

The Release Integrated File System Locks (RLSIFSLCK) command can be used to release all Network File System (NFS) byte-range locks held by a specified NFS client, or to release all byte-range locks (of any type) held on a specified object. This command should only be used to free resources that cannot be freed using normal means.

For more information about byte range locks, see the *fcntl* API in the OS/400 Network File System Support book.

Restriction: You must have *IOSYSCFG special authority to use this command.

Required Parameters

RMTLOCNAME
Specifies the host name or internet address of a remote system whose NFS-related locks on local files are to be released.

To be successful, the remote system name must be valid. You can assign host names to an internet address with the Work with TCP/IP host table entries option on the Configure TCP/IP menu (CFGTCP command). Also, a remote name server can be used to map remote system names to internet addresses. You can use the Change remote name server option on the CFGTCP menu to specify a remote name server.

Host names must follow these conventions:

- The first character must be either A through Z or 0 through 9.
- The last character must be either A through Z or 0 through 9.
- Uppercase and lowercase characters are allowed, but no significance is attached to the case.
- Blanks ( ) are not allowed.
- The special characters, period (.) and minus (-), are allowed.
- Parts of the name separated by periods (.) cannot exceed 63 characters in length.
- Names must be from 1 to 255 characters in length.

’remote-location-name’: Specify the host name or internet address of a remote system whose NFS-related locks on local files are to be released.

OBJ
Specifies the path name of an object on which all byte-range locks are to be released. This will release all locks on that object, regardless of the type of lock or the type of process that is holding them. A message will be sent to each job whose locks are released.

’path-name’: Specify the path name of the local object for which all locks are to be released.

Examples for RLSIFSLCK

Example 1: Releasing Locks for a Remote System

```
RLSIFSLCK   RMTLOCNAME('rainbow1')
```

This command releases the NFS-related locks held on local files by the system named *rainbow1*.

Example 2: Releasing Locks for a Local Object

```
RLSIFSLCK   OBJ('/CustAccounts/May')
```

This command releases all byte-range locks held on the object */CustAccounts/May*.

No error messages.

RLSJOB (Release Job) Command Description

RLSJOB Command syntax diagram

Purpose

The Release Job (RLSJOB) command makes a job eligible for processing after that job has been held from processing by the HLDJOB (Hold Job) command or if the job was submitted to the system as a held job by the JOB or SBMJOB (Submit Job) commands. The job being released could have been on the job queue, output queue, or active in a subsystem (competing for system resources) when it was held. Spooled files that are held because SPLFILE(*YES) is specified in the HLDJOB command are also released. » If the spooled file action (SPLFACN) job attribute is *DETACH and the job is ended while the spooled files are held, the spooled files cannot be released using the Release Job (RLSJOB) command. To release spooled files after the job has been removed from the system, use the Release Spooled File (RLSSPLF) command « .

Restriction: The issuer of the command must be running under a user profile which is the same as the job user identity of the job being released, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority.

The job user identity is the name of the user profile by which a job is known to other jobs. It is described in more detail in the Work Management book.

Required Parameters

JOB Specifies the qualified name of the job being released. If no job qualifier is given, all of the jobs currently in the system are searched for the job name. If more than one of the specified names are found, a qualified job name must be specified.

A job identifier is a qualified name with up to three elements. For example:

```
job-name
user-name/job-name
job-number/user-name/job-name
```

More information on this parameter is in commonly used parameters.

job-name: Specify the name of the job being released.

user-name: Specify the name of the user of the job being released.

job-number: Specify the number of the job being released.

DUPJOBOPT
Specifies the action taken when duplicate jobs are found by this command.

***SELECT:** The selection display is shown when duplicate jobs are found during an interactive session. Otherwise, a message is issued.

***MSG:** A message is issued when duplicate jobs are found.

Examples for RLSJOB

Example 1: Releasing a Job for Processing

```
RLSJOB   JOB(123456)
```

This command releases the job 123456 for processing. If the corresponding HLDJOB command had specified SPLFILE(*YES), any spooled files for job 123456 are also released.

Example 2: Releasing a Job for Processing

```
RLSJOB   JOB(DEPTXYZ/987654)
```

This command releases job name 987654 that was submitted by a user through the user profile DEPTXYZ and later held. The qualified form of the job name is used when jobs with duplicate names exist in the system.

Error messages for RLSJOB

***ESCAPE Messages**

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

CPF1321
Job &1 user &2 job number &3 not found.

CPF1332
End of duplicate job names.

CPF1340
Job control function not performed.

CPF1341
Reader or writer &3/&2/&1 not allowed as job name.

CPF1343
Job &3/&2/&1 not valid job type for function.

CPF1344
Not authorized to control job &3/&2/&1.

CPF1349
Job &3/&2/&1 not released, the job is not held.

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

CPF1352
Function not done. &3/&2/&1 in transition condition.

RLSJOBQ (Release Job Queue) Command Description

RLSJOBQ Command syntax diagram

Purpose

The Release Job Queue (RLSJOBQ) command releases, for additional processing, the jobs on the specified job queue that were previously held by a HLDJOBQ (Hold Job Queue) command. If the jobs were held by something other than a HLDJOBQ command, they are not released.

Required Parameter

JOBQ Specifies the qualified name of the job queue to be released for further processing.

The name of the job queue 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.

job-queue-name: Specify the name of the job queue.

Example for RLSJOBQ

```
RLSJOBQ  JOBQ(QBATCH)
```

Jobs on the job queue QBATCH that were held by a HLDJOBQ command become eligible for processing, including jobs that were placed on the queue while it was being held. Specific jobs that were held by the HLDJOB command or that were put on the job queue in the held state are not released.

Error messages for RLSJOBQ

***ESCAPE Messages**

CPF2207
Not authorized to use object &1 in library &3 type *&2.

CPF3307
Job queue &1 in &2 not found.

CPF3330
Necessary resource not available.

CPF3423
Job queue &1 in library &2 not released. Job queue not held.

RLSJOBSCDE (Release Job Schedule Entry) Command Description

RLSJOBSCDE Command syntax diagram

Purpose

The Release Job Schedule Entry (RLSJOBSCDE) command releases an entry, entries, or generic entries in the job schedule. Each job schedule entry contains the information needed to automatically submit a batch job one time, or at regularly scheduled intervals. If you release a job schedule entry, a job is not submitted immediately, even if the date and time at which it was scheduled to be submitted passed while the entry was held. The job is submitted on any future dates for which the entry is scheduled to be submitted.

Restriction: To release entries, you must have *JOBCTL special authority; otherwise you can release only those entries that you added.

Required Parameters

JOB Specifies the name of the job schedule entry.

***ALL:** All of the job schedule entries for which you have authority are released. If JOB(*ALL) is specified, ENTRYNBR(*ALL) must also be specified.

job-name: Specify the name of the job schedule entry.

generic-job-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. For more information on the use of generic names, refer to generic names. If a generic name is specified, ENTRYNBR(*ALL) must also be specified.

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

***ONLY:** One entry in the job schedule has the job name specified on the JOB parameter. If *ONLY is specified and more than one entry has the specified job name, no entries are released and a message is sent.

***ALL:** All entries with the specified job name are released.

entry-number: Specify the number of the job schedule entry you want to release.

Examples for RLSJOBSCDE

Example 1: Releasing All Job Schedule Entries

```
RLSJOBSCDE  JOB(*ALL)  ENTRYNBR(*ALL)
```

This command releases all the job schedule entries.

Example 2: Releasing an Individual Job Schedule Entry

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

This command releases entry PAYROLL in the job schedule.

Example 3: Releasing a Generic Job Schedule Entry

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

This command releases all entries in the job schedule with the prefix PAY in their names.

Error messages for RLSJOBSCDE

*ESCAPE Messages

CPF1628
Job schedule entry &3 number &4 not found.

CPF1629
Not authorized to job schedule &1.

CPF1630
Not authorized to job schedule entry &3 number &4.

CPF1632
Job schedule entry &3 number &4 damaged.

CPF1636
More than one entry with specified entry job name found.

CPF1637
Job schedule &1 in library &2 in use.

CPF1638
Job schedule entry &3 number &4 in use.

CPF1640
Job schedule &1 in library &2 does not exist.

CPF1641
Job schedule &1 in library &2 damaged.

CPF1645
No job schedule entries found for specified name.

CPF1646
Entry number must be *ALL when generic name specified.

CPF1648
&3 entries successfully released. &4 entries not released.

CPF1649
Entry number must be *ALL.

RLSJOBJS (Release Job using Job Scheduler) Command Description

Note: To use this command, you must have the 5722-JS1 (Job Scheduler for iSeries) licensed program installed.

RLSJOBJS Command syntax diagram

Purpose

The Release Job using Job Scheduler (RLSJOBJS) command allows you to release a job that you specify.

Note: When referring to a job in this command, we are referring to an entry in Job Scheduler.

An **entry** in Job Scheduler is a user-defined name for commands or programs that you want to process at scheduled times and dates. Job Scheduler jobs (entries) are not OS/400 objects.

When you press Enter, a message is displayed confirming that the job you selected has been released. The job is released until the Hold Job using Job Scheduler (HLDJOBJS) command is processed for the job or Option 3 (Hold) is selected for the job on the Work with Jobs display.

Required Parameter

JOB Specifies the name of the job that you want to release.

You must specify a job and optionally can specify a group to which the job belongs and the associated sequence number of the job.

Element 1: Job

job-name: Specify the name of the job that you want to release.

Element 2: Group

***NONE:** The job is not a member of a group.

group-name: Specify the name of the group to which the job belongs.

Element 3: Group sequence

***NONE:** The job does not have a sequence number.

group-sequence-number: Specify the sequence number of the job in the group.

Example for RLSJOBJS

Example 1: Releasing a Job that has been Held

```
RLSJOBJS JOB(JOB02)
```

In this example JOB02 is released, having been previously held using a HLDJOBJS command.

Error messages for RLSJOBJS

None

RLSOUTQ (Release Output Queue) Command Description

RLSOUTQ Command syntax diagram

Purpose

The Release Output Queue (RLSOUTQ) command releases an output queue that was previously held by a HLDOUTQ (Hold Output Queue) command. This command allows all currently waiting spooled files, and all spooled files that are added to the output queue after the command is sent, to be processed by a spooling writer. Files held by a HLDSPLF (Hold Spooled File) command or created in a held state, are not released.

Required Parameter

OUTQ Specifies the qualified name of the output queue.

The name of the output queue 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.

output-queue-name: Specify the name of the output queue being released.

Example for RLSOUTQ

```
RLSOUTQ  OUTQ(PRINTER)
```

On the output queue named PRINTER, spooled files that were held by a HLDOUTQ command are released for further processing. This includes spooled files placed on the queue while it was being held, except for specific files that have been held by the HLDSPLF command or were put on the queue in hold.

Error messages for RLSOUTQ

***ESCAPE Messages**

CPF2207
Not authorized to use object &1 in library &3 type *&2.

CPF3330
Necessary resource not available.

CPF3357
Output queue &1 in library &2 not found.

CPF3424
Output queue &1 in library &2 not released. Output queue not held.

RLSPTF (Release Program Temporary Fix) Command Description

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

RLSPTF Command syntax diagram

Purpose

The Release Program Temporary Fix (RLSPTF) command releases a program temporary fix (PTF) save file (SAVF) that was previously on hold. When a PTF SAVF is released, it can be automatically sent to a service requester by way of electronic customer support.

Required Parameter

PTF Specifies which PTF is to be released.

Optional Parameters

LICPGM
Specifies the product identifier (ID) of the PTF being released.

***ALL:** All products are searched for the PTF being released.

product-ID: Specify the product ID of the PTF being released.

RLS Specifies the release level of the PTF to be released.

***ONLY:** The PTF exists for only one release level of the product.

version-release-modification: Specify the release level of the PTF to be released.

Example for RLSPTF

```
RLSPTF  PTF(SI12345)
```

This command releases the save files of the PTF named SI12345. Releasing SI12345 allows it to be automatically distributed to a service requester.

Error messages for RLSPTF

***ESCAPE Messages**

SMU1430
Duplicate PTF found.

«

RLSRDR (Release Reader) Command Description

RLSRDR Command syntax diagram

Purpose

The Release Reader (RLSRDR) command releases the specified spooling reader making it available to again process jobs on the job queue. The specified reader was held by a previous HLDRDR (Hold Reader) command. Data is not lost.

Required Parameter

RDR Specifies the name of the spooling reader being released. Specify the name of the spooling reader.

Example for RLSRDR

```
RLSRDR  RDR(DISKETTE)
```

This command releases the diskette reader named DISKETTE for additional processing.

Error messages for RLSRDR

***ESCAPE Messages**

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

CPF1340
Job control function not performed.

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

CPF1352
Function not done. &3/&2/&1 in transition condition.

CPF3312
Reader &1 neither active nor on job queue.

CPF3315
Reader &3/&2/&1 not released. Reader not held.

CPF3330
Necessary resource not available.

CPF3490
Not authorized to specified reader.

RLSRMTPHS (Release Remote Phase) Command Description

RLSRMTPHS Command syntax diagram

Purpose

The Release Remote Phase (RLSRMTPHS) command initiates a session between the iSeries 400 and a System/370* NetView* Distribution Manager (NDM) host system. After the NDM releases the phase, or there is an unsuccessful attempt to release the phase, the session is ended.

The following considerations apply when running this command:

- The NDM phase specified by the PHASE parameter must exist and be part of the NDM plan specified by the PLAN parameter.
- The NDM phase specified by the PHASE parameter must be in a HELD state on the host system.
- The NDM plan specified by the PLAN parameter must exist and have been previously submitted to the NDM host application specified by the APPID parameter.
- The device specified by the DEV parameter must be a Systems Network Architecture upline facility (SNUF) device and must be program start request (PSR) capable.
- This command runs only on a node which is currently functioning as a host interface node to the NDM host system. However, it is not restricted to releasing only those NDM phases whose destination is the node sending the command. Any phase may be released for any node that shares the same host interface node sending this command.

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

Required Parameters

PHASE
Specifies the name of the NetView Distribution Manager phase that is released. This phase must exist on the NDM host system as part of the plan specified by the PLAN parameter, and must be in a HELD state.

PLAN Specifies the name of the NetView Distribution Manager plan that contains the phase that is released. This plan must exist on the NDM host.

APPID
Specifies the name of the NetView Distribution Manager application under which the plan name specified by the PLAN parameter was submitted. This is the same name by which NDM was made known to MVS when it was generated.

RMTLOCNAME
Specifies the remote location name of the system with which this object communicates.

Note: The device with which the user's program is communicating is specified on the DEV parameter.

DEV Specifies the device name of the iSeries 400 device that is used for the communications session started as a result of this command. The device must be a SNUF device and must be PSR capable.

Example for RLSRMTPHS

```
RLSRMTPHS  PHASE(MESSAGE)  PLAN(ALEXPLAN)  APPID(DSXNDM)
  RMTLOCNAME(A083187)  DEV(SNUFDEV)
```

This command initiates a session using device SNUFDEV with remote location name A083187 to connect with the System/370 NetView Distribution Manager host application DSXNDM. After the session connection is made, phase MESSAGE, as part of plan ALEXPLAN, attempts to release. If the release is successful, message CPC8889 (Phase MESSAGE released by NetView Distribution Manager) is sent. If the release is not successful, message CPF8880 (Phase MESSAGE not released by Netview Distribution Manager) is sent.

Error messages for RLSRMTPHS

*ESCAPE Messages

CPF8880
Phase &1 not released by NetView Distribution Manager.

RLSSPLF (Release Spooled File) Command Description

RLSSPLF Command syntax diagram

Purpose

The Release Spooled File (RLSSPLF) command releases a specified spooled file formerly held on an output queue for processing by a spooling writer. The RLSSPLF command can release a spooled file that was held by:

- A HLDSPLF command
- HOLD(*YES) being specified in the device file or on an override command
- SAVE(*YES) being specified in the device file, on an override command, or in the CHGSPLFA command
- A HLDWTR command and a RLSWTR command with OPTION(*BYPASS) specified
- The operator canceling a system request to put forms on the printer
- » HLDJOB SPLF(*YES) if the job is later ended while the job is held and the spooled file action (SPLFACN) job attribute is *DETACH.«

Required Parameter

FILE Specifies the name of the spooled file that is being released to be written to an output device.

***SELECT:** All spooled files that meet the selection requirements specified in the SELECT keyword are released. This value is mutually exclusive with the » JOB, SPLNBR, JOBSYSNAME, and CRTDATE parameters « . Specifying *SELECT causes the » JOB, SPLNBR, JOBSYSNAME, and CRTDATE parameters « to be ignored.

spooled-file-name: Specify the name of the spooled file being released.

Optional Parameters

JOB Specifies the name of the job that created the file being released for additional processing.

A job identifier is a special value or a qualified name with up to three elements. For example:

```
*
job-name
user-name/job-name
job-number/user-name/job-name
```

More information on this parameter is in commonly used parameters.

***:** The job that issued this RLSSPLF command is the job that produced the spooled file being released. If no job qualifier is given, all of the jobs currently in the system are searched for the simple job name.

job-name: Specify the name of the job that created the spooled file.

user-name: Specify the name of the user of the job that created the file being released.

job-number: Specify the number of the job that created the file being released.

SPLNBR

Specifies the number of the spooled file being released. More information on this parameter is in commonly used parameters.

***ONLY:** One spooled file from the job has the specified file name. The number of the spooled file is not necessary. If *ONLY is specified and more than one spooled file has the specified file name, a message is sent.

***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 creation 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 to release that has the specified file 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 creation date and time.

***CURRENT:** The spooled file created on the current system with the specified job name, user name, job number, spooled file name, and spooled file number is used.

***ANY:** The job system name is not used to determine which spooled file is used. Use this value when the spooled file creation date and time parameter is to take precedence over the job system name when selecting a spooled file.

system name: Specify the name of the system where the job that created the spooled file ran.

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.

***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 creation 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: Date spooled file was created

date: Specify the date the spooled file was created.

Element 2: Time spooled file was created

***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 creation date.

***LAST:** The spooled file with the latest creation time of the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file creation date is used.

time: Specify the time the spooled file was created. «

SELECT

Specifies which group of files is selected for being released. Files can be selected based on user, device, form type, or user data. Only files that meet each of the requirements are selected.

Element 1: User Values

***CURRENT:** Only files created by the user running this command are released.

***ALL:** Files created by all users are released.

user-name: Specify the user name of the files being released.

Element 2: Device Values

***ALL:** Files queued for any device or on any object queue are released.

***OUTQ:** All files that are not queued for a device are released. These files are on output queues that are not associated with printers.

device-name: Specify the name of the device whose queued files are released.

Element 3: Form Type Values

***ALL:** Files for all form types are released.

***STD:** Only files that specify the standard form type are selected.

form-type: Only files with the specified form type are released.

Element 4: User Data Values

***ALL:** Files with any user data tag specified are released.

user-data: Only files with the specified user data tag are released.

Example for RLSSPLF

```
RLSSPLF  FILE(STOCK14)  JOB(000047/SMITH/MASTER)
```

This command releases the spooled file named STOCK14 created in the job named MASTER. The file can now be selected for processing by the spooling writer. The job was run under the user profile named SMITH and was assigned the job number 000047 by the system.

Error messages for RLSSPLF

***ESCAPE Messages**

CPF33D0

Printer &1 does not exist.

CPF33D1

User &1 does not exist.

CPF3303

File &1 not found in job &5/&4/&3.

CPF3304

File &1 number &2 cannot be released.

CPF3309

No files named &1 are active.

CPF3322

File &1 number &2 not released.

CPF3330

Necessary resource not available.

CPF3340
More than one file with specified name found in job &5/&4/&3.

CPF3342
Job &5/&4/&3 not found.

CPF3343
Duplicate job names found.

CPF3344
File &1 number &2 no longer in the system.

CPF3357
Output queue &1 in library &2 not found.

CPF3362
Objects in QTEMP not valid for parameter values.

CPF3492
Not authorized to spooled file.

»

RLSSBMCRQA (Release Submitted Change Request Activity) Command Description

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

RLSSBMCRQA Command syntax diagram

Purpose

The Release Submitted Change Request Activity (RLSSBMCRQA) command releases one or more change request activities.

Restrictions:

1. Only activities in the Held or Not Started status can be released.
2. You must be either the submitter of the change request or have *JOBCTL special authority.

Required Parameter

CRQ Specifies the change request name and the change request sequence number whose activities are to be released.

Element 1: Change Request Name

change-request-name: Specify the name of the change request whose activities are to be released.

Element 2: Sequence Number

sequence-number: Specify the change request sequence number of the activities that are to be released.

Optional Parameters

ACTIVITY
Specifies the activity to be released.

ALL: Release all activities of the specified change request.

***LAST:** Release the activity named *LAST. This is not the last activity added to the change request description object. This is the last activity that runs after the change request is submitted.

activity-name: Specify the name of the activity to be released.

CPNAME

Specifies the destination node for an activity that should be released. The node consists of two elements that are the network identifier and the control point name.

Element 1: Network Identifier

***ALL:** All the activities for the change request specified are released regardless of the control point name of the managed system on which the activity is to be performed.

***NETATR:** Only activities having network IDs that match the one defined in the network attributes for this system are released.

network-identifier: Specify a network ID. Only the activities having the network ID and control point name specified are released.

Element 2: Control Point Name

***ALL:** All the activities for the change request specified are released regardless of the control point name of the managed system on which the activity is to be performed.

***NETATR:** Only activities for the control point name matching the one defined in the network attributes for this system are released.

control-point-name: Specify a control point name. Only activities having the network ID and the control point name specified are released. For NetView Distribution Management Agents, the control point name is the change control client which supports numeric characters (0-9) in the first position of control point names that are valid in other platforms.

Example for RLSSBMCRQA

```
RLSSBMCRQA CRQ(CHG001 456) ACTIVITY(*ALL) CPNAME(*ALL)
```

Releases all of the activities for the change request named CHG001 with sequence number 456 for all the nodes.

Error messages for RLSSBMCRQA

***ESCAPE Messages**

None «

RLSWTR (Release Writer) Command Description

RLSWTR Command syntax diagram

Purpose

The Release Writer (RLSWTR) command releases a held writer for additional processing. The specified writer was held by a previously issued HLDWTR (Hold Writer) command. If the writer was writing a file when it was held, the writer can be released either to resume writing the file it was writing or to start writing the next file. In any case, data from the file that was being written when the HLDWTR command was issued is not lost.

Required Parameter

WTR Specifies the name of the spooling writer being released to do additional processing.

Optional Parameters

OPTION
Specifies the point in the file where the writer is released. Also, a diskette writer can be released if *CURRENT is specified.

***CURRENT:** The writer is released at the point where it was held by the HLDWTR (Hold Writer) command. This option may be specified when the writer is not producing a file.

***BEGIN:** The writer is released at the beginning of the current file. This option may be specified only if the writer was held while producing the current file.

***BYPASS:** The writer is released at the beginning of the next file. The current file is implicitly held on the queue. This option may be specified only if the writer was held while producing the current file.

+number: Specify the number of pages past the point where the writer was held that it is released. An error message is sent to the user if the specified value goes beyond the number of records or printed pages in the file. This option may be specified only if the writer was held while producing the file.

-number: Specify the number of pages preceding the point where the writer was held that it is released. An error message is sent to the user if the specified value is greater than the number of records or printed pages previously processed in the file before the writer was held. This option may be specified only if the writer was held while producing the file.

PAGE Specifies the page where the writer starts printing. This parameter is mutually exclusive with the OPTION parameter and is only valid for a print writer. An error message is sent if the PAGE parameter is specified for a diskette writer. Specify the page number where the writer starts printing.

Examples for RLSWTR

Example 1: Releasing a Writer at Beginning of File

```
RLSWTR  WTR(PRINTER)  OPTION(*BEGIN)
```

This command releases writer PRINTER to begin producing the current file at its beginning.

Example 2: Releasing Writer at Specified Point

```
RLSWTR  WTR(PRTR)  OPTION(-3)
```

This command releases writer PRTR to begin printing again at a point three pages before the point where the writer was held. That is, the last three pages previously printed are the first three pages printed this time.

Example 3: Starting Printing on Page Ten

```
RLSWTR  WTR(PRTR)  PAGE(10)
```

This command releases writer PRTR to start printing again at page ten.

Error messages for RLSWTR

*ESCAPE Messages

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

CPF1340
Job control function not performed.

CPF1352
Function not done. &3/&2/&1 in transition condition.

CPF3313
Writer &1 not active nor on job queue.

CPF3314
PAGE parameter allowed only for print writers.

CPF3316
Writer &3/&2/&1 not released because writer not held.

CPF3317
OPTION parameter value not allowed for diskette writer.

CPF3330
Necessary resource not available.

CPF3331
Not authorized to control writer &3/&2/&1.

CPF3334
Previous hold to writer &3/&2/&1 pending.

RMVACC (Remove Access Code) Command Description

RMVACC Command syntax diagram

Purpose

The Remove Access Code (RMVACC) command removes an access code from the system that was previously defined by the Add Access Code (ADDACC) command.

Notes:

1. This command can take a long time to run because it must update each object in the document library that has been assigned the access code being removed.
2. This command removes the access code from all filed documents and folders, from all users authorized to the access code, and from the system.

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

Required Parameter

ACC Specifies the access codes to remove from the system. The access code is a decimal number ranging from 1 through 2047. If the access code specified is not defined to the system, a diagnostic message is sent, and processing continues with any additional access codes specified.

Example for RMVACC

```
RMVACC ACC(300)
```

This command removes access code 300 from the system.

Error messages for RMVACC

***ESCAPE Messages**

CPF90A5
Access codes could not be removed.

CPF9009
System requires file &1 in &2 be journaled.

CPF9011
&1 access codes removed, &2 not removed.

CPF9024
System cannot get correct record to finish operation.

CPF9845
Error occurred while opening file &1.

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

CPF9847
Error occurred while closing file &1 in library &2.

RMVALRD (Remove Alert Description) Command Description

RMVALRD Command syntax diagram

Purpose

The Remove Alert Description (RMVALRD) command allows you to remove an alert description that was added previously by the ADDALRD command. More information on alerts is in the Alerts Support book.

Required Parameters

MSGID
Specifies the message identifier for the alert description to be removed.

ALRTBL
Specifies the alert table from which this alert description is removed.

The name of the alert table can be qualified by one of the following library values:

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

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

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

alert-table-name: Specify the name of the alert table.

Example for RMVALRD

```
RMVALRD  MSGID(USR1234)  ALRTBL(USER/USRMSGS)
```

This command removes the alert description for message identifier USR1234.

Error messages for RMVALRD

***ESCAPE Messages**

CPF1A04
Alert code &1 not found in alert table &2.

CPF1A05
Alert table &1 in &2 damaged.

CPF2499
Message identifier &1 not allowed.

CPF7BB1
Alert description not found.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9811
Program &1 in library &2 not found.

CPF9812
File &1 in library &2 not found.

CPF9814
Device &1 not found.

CPF9820
Not authorized to use library &1.

CPF9821
Not authorized to program &1 in library &2.

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

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9831
Cannot assign device &1.

RMVAUTLE (Remove Authorization List Entry) Command Description

RMVAUTLE Command syntax diagram

Purpose

The Remove Authorization List Entry (RMVAUTLE) command removes user entries from an authorization list. The authorization list must already exist.

Restrictions:

1. Only the owner of the authorization list, a user with authorization list management authority (*AUTLMGT) on the authorization list, or a user with all object (*ALLOBJ) authority can use this command.
2. The user with *AUTLMGT authority can only remove a user if the user with *AUTLMGT authority has at least the same specific authorities as the user being removed.

Required Parameters

AUTL Specifies the name of the authorization list from which the user entries are removed. The authorization list must exist.

authorization-list-name: Specify the authorization list name from which the user entries are removed.

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

USER Specifies a list of user names to be removed from the authorization list. Up to 50 user entries can be specified.

Example for RMVAUTLE

```
RMVAUTLE   AUTL(PAYROLL)  USER(TOM JULIE KAREN)
```

This command removes users TOM, JULIE, and KAREN from the authorization list PAYROLL.

Error messages for RMVAUTLE

***ESCAPE Messages**

CPF2253
No objects found for &1 in library &2.

CPF2281
The users specified do not exist on the system.

CPF2283
Authorization list &1 does not exist.

CPF2284
Not authorized to change authorization list &1.

CPF2285
&1 errors removing users, &2 authorization lists processed.

CPF2288
*PUBLIC cannot be removed from an authorization list.

CPF2289
Unable to allocate authorization list &1.

RMVAJE (Remove Autostart Job Entry) Command Description

RMVAJE Command syntax diagram

Purpose

The Remove Autostart Job Entry (RMVAJE) command removes a job entry that starts automatically from the specified subsystem description.

Restriction: To use this command, the user must have object operational and object management authorities for the specified subsystem description.

Required Parameters

SBSD Specifies the qualified name of the subsystem description from which the autostart job entry is being removed.

The name of the subsystem description 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.

subsystem-description-name: Specify the name of the subsystem description from which the autostart job entry is being removed.

JOB Specifies the name of the job entry to remove.

Example for RMVAJE

```
RMVAJE  SBSD(MYLIB/PAYROLL)  JOB(INITIAL)
```

This command removes job entry named INITIAL that starts automatically from the PAYROLL subsystem description in the library MYLIB.

Error messages for RMVAJE

***ESCAPE Messages**

CPF1619
Subsystem description &1 in library &2 damaged.

CPF1697
Subsystem description &1 not changed.

RMVBNDDIRE (Remove Binding Directory Entry) Command Description

RMVBNDDIRE Command syntax diagram

Purpose

The Remove Binding Directory Entry (RMVBNDDIRE) command removes an entry from the binding directory.

Restrictions:

1. You must have *READ and *OBJOPR authority for the library where the binding directory is being updated.
2. You must have *OBJOPR and *DELETE authority to the binding directory.

Required Parameter

BNDDIR
Specifies the binding directory from which an entry is removed.

The name of the binding directory can be qualified by one of the following library values:

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

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

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

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

binding-directory-name: Specify the name of the binding directory to be updated.

Optional Parameter

OBJ Specifies the object name to be removed from the binding directory. If no objects are specified, zero entries are removed from the binding directory.

Element 1: Object Name

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

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

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

***ALL:** All objects with the specified type are removed from the specified library.

object-name: Specify the name of the object to remove.

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

Element 2: Object Type

***SRVPGM:** Indicates the object being removed is a service program.

***MODULE:** Indicates the object being removed is a module.

Example for RMVBNDDIRE

```
RMVBNDDIRE   BNDDIR(SOURCE)  OBJ(LIST)
```

This command allows you to remove the object LIST from the binding directory SOURCE.

Error messages for RMVBNDDIRE

***ESCAPE Messages**

CPF5D01
Binding directory &1 in library &2 is not usable.

CPF5D09
Object &2/&1 type &3 was not found in binding directory &4 in library &5.

CPF980F
Binding directory &1 in library &2 not found.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9830
Cannot assign library &1.

RMVBKP (Remove Breakpoint) Command Description

RMVBKP Command syntax diagram

Purpose

The Remove Breakpoint (RMVBKP) command removes one or more breakpoints from the specified program being debugged. It can also remove all breakpoints from all programs in debug mode.

Restrictions:

1. This command is valid only in debug mode. To start debug mode, refer to the STRDBG (Start Debug) command.
2. This command cannot be used if the user is servicing another job, and that job is on a job queue, or is being held, suspended, or ended.
3. This command cannot be used to remove breakpoints from a bound program.

Optional Parameters

STMT Specifies which high-level language (HLL) statements or machine instructions in a program have their breakpoints removed. Breakpoints can be removed from a specified program (PGM parameter) or from the most recent program that has reached a breakpoint (STMT(*) specified). If a program is specified, one or more statement identifiers can be specified or all the breakpoints can be specified. If STMT(*) is specified, the breakpoint that the most recently stopped program has reached is removed. Also, all breakpoints can be removed from all programs in debug mode.

***:** The most recent breakpoint at which a program is currently stopped is the breakpoint removed.

***ALL:** All breakpoints in the specified program are removed.

statement-identifier: Specify the statement identifiers removed from the program specified by the PGM parameter. No more than 10 identifiers can be specified.

PGM Specifies the program from which the specified breakpoints are removed. This parameter can be specified only if STMT (*) is omitted.

***DFTPGM:** The default program is the program whose breakpoints are removed.

***ALL:** All programs currently in debug mode have their breakpoints removed. PGM(*ALL) is valid only if the STMT parameter is not specified.

program-name: Specify the name of the program from which the specified breakpoints are removed.

Example for RMVBKP

```
RMVBKP   STMT(100)
```

This command removes the breakpoint that is on statement 100 from the default program.

Error messages for RMVBKP

***ESCAPE Messages**

CPF1999
Errors occurred on command.

»

RMVCCSCLT (Remove Change Control Server Client) Command Description

Note: To use this command, you must have the 5722-MG1 (Managed System Services for iSeries) licensed program installed.

RMVCCSCLT Command syntax diagram

Purpose

The Remove Change Control Server Client (RMVCCSCLT) command allows the user to remove a client defined in the change control server database. The asterisk (*), the slash (/), and embedded blanks are not allowed.

Required Parameter

CLIENT
Specifies the name of the client in the change control server dtaabase that is being removed.

client-name: Specify a maximum of 63 characters for the name of the client. The name of the client must be precisely specified. If you specify the name of a non existing client, the command is rejected.

Another alternative when you define a client is to use the automatic target registration attribute. You can specify this attribute with the Change CC Server Attribute (CHGCCSA) command. With the Automatic target registration value, the client is automatically defined the first time it connects to the change control server, and you do not need to define the client then.

Example for RMVCCSCLT

Example 1: Remove client CONNERY

```
RMVCCSCLT CLIENT(CONNERY)
```

Error messages for RMVCCSCLT

<u>*ESCAPE Messages</u>

MSS0123
Internal processing error occurred.

MSS0A2F
Client name not found.

MSS0A30
Pending request for this client.

MSS0A40
Change control client name not valid.

MSS0A41
Change control client name cannot be removed.

MSS2109
Change control client not removed.

«»

RMVCRQDA (Remove Change Request Description Activity) Command Description

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

RMVCRQDA Command syntax diagram

Purpose

The Remove Change Request Description Activity (RMVCRQDA) command removes an activity from a change request description.

Restriction: You must have *CHANGE authority to the change request description object to remove an activity from it.

Required Parameters

CRQD Specifies the name and library of the change request description where the activity is removed.

The possible library values are:

***LIBL:** Locates the change request description in the library list. The first matching change request description is used.

***CURLIB:** Locates the change request description in the current library. If no library is specified as the job's current library, the QGPL library is used.

library-name: Specify that only the library specified is searched.

change-request-description-name: Specify the name of the change requests description from which an activity is removed.

ACTIVITY

Specifies the activity to remove.

***LAST:** The last activity to run (named *LAST) is removed.

activity-name: Specify the name of the activity to be removed.

Example for RMVCRQDA

```
RMVCRQDA CRQD(*LIBL/CHG001)  ACTIVITY(ACT01)
```

This command removes the ACT01 activity from the first change request description found in the library list of the current interactive job with the name CHG001.

Error messages for RMVCRQDA

***ESCAPE Messages**

None «»

RMVCLUNODE (Remove Cluster Node Entry) Command Description

RMVCLUNODE Command syntax diagram

Purpose

The Remove Cluster Node Entry (RMVCLUNODE) command is used to remove a node from a cluster. The node specified will be removed from the cluster membership list and will no longer be considered a member of the cluster. The node will also be removed from the membership of the device domain to which it belongs. The cluster resource group objects on the node being removed are deleted only if the node has a status of Active or if this command is called on the node that is being removed.

A node can be removed regardless of its status. If this command is called on a node with a status of Active, any node in the cluster can be removed. If this command is called on a node with a status of Inactive or Failed, only the node running the command can be removed. To remove a node that is not active, this command should be called on the node being removed and on a node in the cluster that is active.

If the node being removed is active, the cluster resource group exit program will be passed an action code of Remove Node. The exit program on all other nodes in the recovery domain of the cluster resource group will be passed an action code of Failover. If the node being removed is the primary node for a device cluster resource group, ownership of the hardware associated with the cluster resource group will be moved to a backup node. If there are no backup nodes or all the backup nodes are either inactive or in a different cluster partition, ownership of the hardware is left with the node being removed.

If the node being removed is inactive, the cluster resource group exit program will be passed an action code of Rremove Node on all nodes in the recovery domain. Ownership of hardware associated with a device cluster resource group will not be changed but will remain with the node being removed. If the node being removed is inactive, the cluster resource group exit program will be called with an action code of Delete Command on the node being removed if the command is run on the node being removed.

If the node being removed is a member of a device domain and it later will be added back to a cluster, the node most likely will need to be IPLed before it can be added to any device domain. One example of this situation would be if a device description for an auxiliary storage pool has been varied on since the last IPL.

Restrictions

1. To use this command you must have *IOSYSCFG authority.
2. This CL command cannot be called from a cluster resource group exit program.
3. If all of the nodes in the cluster have a status of New, this command can only be called on the node where the cluster was originally created.
4. There must be more than one node in the membership list.

Required Parameters

CLUSTER
Specifies the name of the cluster that contains the node being removed.

cluster-name: Specify the name of the cluster.

NODE Specifies the node identifier being removed.

node-identifier: Specify the name of the node being removed.

Example for RMVCLUNODE

```
RMVCLUNODE CLUSTER(MYCLUSTER) NODE(RCHCST01)
```

This command removes the node RCHCST01 from the cluster membership for cluster MYCLUSTER. Cluster Resource Services is ended on node RCHCST01.

Error messages for RMVCLUNODE

*ESCAPE Messages

CPF0001
Error found on &1 command.

«»

RMVCRGDEVE (Remove Cluster Resource Group Device Entry) Command Description

RMVCRGDEVE Command syntax diagram

Purpose

The Remove Cluster Resource Group Device Entry (RMVCRGDEVE) command removes one or more configuration objects from a device cluster resource group. All configuration object entries can be removed but at least one configuration object entry must exist before the Start Cluster Resource Group (STRCRG) command can be called.

Ownership of the hardware associated with the configuration object being removed is not affected. The hardware is still owned by whatever node owned it before this command was called.

If an exit program is specified for the cluster resource group, the cluster resource group exit program is called with an action code of Remove Device Entry on all active nodes in the recovery domain. The cluster resource group status is set to Remove Device Entry Pending. If the exit program completes successfully, the cluster resource group status is reset to its value at the time the command was called. If the exit program fails and the cluster resource group cannot be restored to its original condition, the cluster resource group status is set to Indoubt.

Restrictions

1. To use this command you must have *IOSYSCFG authority.
2. This CL command cannot be called from a cluster resource group exit program.
3. Cluster Resource Services must be active on the node processing the request.
4. At least one node in the recovery domain must be active.
5. If the cluster resource group is Active, the last device entry cannot be removed. The cluster resource group must be ended first.

Required Parameters

CLUSTER
Specifies the name of the cluster to which the cluster resource group belongs.

cluster-name: Specify the name of the cluster which contains the cluster resource group.

CRG Specifies the name of the cluster resource group from which device list entries are being removed.

cluster-resource-group-name: Specify the name of the cluster resource group.

CFGOBJ
Specifies the resilient devices that are to be removed from the cluster resource group.

Element 1: Configuration Object Name

Specifies the auxiliary storage pool device description object which will be removed from the device list of the cluster resource group.

configuration-object: Specify the device description name.

Element 2: Configuration Object Type

Specifies the type of configuration object specified with configuration object name.

***DEVD:** Type of configuration object is a device description.

Example for RMVCRGDEVE

```
RMVCRGDEVE CLUSTER(MYCLUSTER) CRG(MYCRG) CFGOBJ((IASP01 *DEVD))
```

This command removes configuration object IASP01 from an existing cluster resource group MYCRG in cluster MYCLUSTER.

Error messages for RMVCRGDEVE

***ESCAPE Messages**

CPF0001
Error found on &1 command.

«»

RMVCRGNODE (Remove Cluster Resource Group Node Entry) Command Description

RMVCRGNODE Command syntax diagram

Purpose

The Remove Cluster Resource Group Node Entry (RMVCRGNODE) command is used to remove a node from the recovery domain of a cluster resource group. The node being removed does not need to be active in the cluster to be removed from the recovery domain. When the node is removed from the recovery domain, the cluster resource group object is deleted from that system. If the cluster resource group is active, the primary node may **not** be removed from the cluster resouce group. If the cluster resource group has no backup nodes, the primary node cannot be removed.

This command will do the following for all cluster resource group types:

- Set the cluster resource group status to Remove Node Pending.
- Call the exit program on all active nodes in the recovery domain with an action code of REMOVE NODE, if an exit program is specified for the cluster resource group.
- Reset the cluster resource group status to the value at the time the command was called, if the exit program completes successfully on all nodes.
- Set the cluster resource group status to Indoubt if the exit program fails on any node and the original state of the cluster resource group cannot be recovered.

This command will do the following for application cluster resource groups:

- If Cluster Resource Services configured the takeover IP address, it will be removed.

This command will do the following for device cluster resource groups:

- If the node being removed is the current primary node, ownership of the devices specified in the cluster resource group are switched from the current primary to the new primary if none of the configuration objects are varied on on the current primary. If any configuration objects are varied on, an error message is returned. In addition, the new primary node must be active. Devices are not varied on after the ownership is switched. The node which is to become the new primary must be active in the cluster.

Restrictions

1. This CL command cannot be called from a cluster resource group exit program.
2. Cluster Resource Services must be started on the node running the command.
3. The cluster resource group status must be other than active in order to remove the node that is currently the primary.
4. At least one backup node must exist in the recovery domain of the cluster resource group, if the primary node is removed.

5. At least node must remain active in the recovery domain of the cluster resource group after the successful completion of the remove operation.

Required Parameter

CLUSTER
Specifies the name of the cluster containing the cluster resource group.

cluster-name: Specify the name of the cluster.

CRG Specifies the name of the cluster resource group that is to have a node removed from its recovery domain.

cluster-resource-group-name: Specify the name of the cluster resource group from which the node will be removed.

NODE Specifies the name of the node that is to be removed from the recovery domain of the cluster resource group.

node-identifier: Specify the name of the node that is to be removed.

Example for RMVCRGNODE

```
RMVCRGNODE CLUSTER(MYCLUSTER) CRG(MYCRG) NODE(RCHCST03)
```

This command removes node RCHCST03 from the recovery domain of cluster resource group MYCRG in cluster MYCLUSTER.

Error messages for RMVCRGNODE

***ESCAPE Messages**

CPF0001
Error found on &1 command.

«

RMVCMNE (Remove Communications Entry) Command Description

RMVCMNE Command syntax diagram

Purpose

The Remove Communications Entry (RMVCMNE) command removes a communications device entry from an existing subsystem description.

Restrictions:

1. The user of this command must have object operational and object management authorities for the specified subsystem description.
2. All jobs that are active through the communications entry being removed must be ended before this command can be run.
3. A communications entry cannot be removed from an active subsystem while there are active jobs associated with it.

Required Parameters

SBSD Specifies the qualified name of the subsystem description from which the communications device entry is being removed.

The name of the subsystem description 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.

subsystem-description-name: Specify the name of the subsystem description.

DEV Specifies the name of the device description or the type of the device for which a communications device entry is being removed from the subsystem description.

Note: The DEV parameter and the RMTLOCNAME parameter are mutually exclusive.

***ALL:** The *ALL communication entry is removed.

***APPC:** The advanced program-to-program communications device entry is removed.

***ASYNC:** The asynchronous communications device entry is removed with this communications entry. This value is valid only when MODE(*ANY) is specified.

***BSCEL:** The binary synchronous equivalency link communications device entry is removed. This value is valid only when MODE(*ANY) is specified.

***FINANCE:** The FINANCE communications device entry is removed. This value is valid only when MODE(*ANY) is specified.

***INTRA:** The INTRA communications device entry is removed. This value is valid only when MODE(*ANY) is specified.

***RETAIL:** The RETAIL communications device entry is removed. This value is valid only when MODE(*ANY) is specified.

***SNUF:** The SNA upline facility communications device entry is removed. This value is valid only when MODE(*ANY) is specified.

device-name: Specify the device description name or the type of device to use with this communications device entry. The name specified on the CRTDEVxxx command associated with this device description name is used.

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

RMTLOCNAME
Specifies the name of the remote location that is used with this object.

Note: The remote location name specified on the CRTDEV command can be used. Validity checking is not done on the remote location name.

Optional Parameters

MODE Specifies the name of the mode of the device specified by the DEV parameter or the remote location specified by the RMTLOCNAME parameter from which the communications device entry is removed.

***ANY:** The communications device entry which has a value of *ANY for the device is removed.

mode-name: Specify the name of the mode entry of the communications device or remote location from which the communications device entry is removed.

Example for RMVCMNE

```
RMVCMNE  SBSD(LIB2/SBS1)  DEV(COMDEV)
```

This command removes the communications device entry for the device COMDEV from the subsystem description SBS1 in library LIB2.

Error messages for RMVCMNE

***ESCAPE Messages**

CPF1619
Subsystem description &1 in library &2 damaged.

CPF1691
Active subsystem description may or may not have changed.

CPF1697
Subsystem description &1 not changed.

RMVCOMSNMP (Remove Community for SNMP) Command Description

RMVCOMSNMP Command syntax diagram

Purpose

The Remove Community for SNMP (RMVCOMSNMP) command is used to remove a Simple Network Management Protocol (SNMP) community profile from the SNMP agent community list. The community profile consists of a community name, an object access specification, and a list of the SNMP managers that are part of the community. The combination of the community name (COM) and the translate to ASCII community (ASCIICOM) parameters defines a community.

Required Parameter

COM Specifies the name of the SNMP community being removed. The community must already exist in the SNMP agent community list.

community-name: Specify the name of the SNMP community being removed. The name may contain characters that cannot be displayed.

Optional Parameter

ASCIICOM
Specifies whether the community name is translated to ASCII characters before it is compared with the community name specified in a request from an SNMP manager. This parameter is used

in combination with the community name to determine the community to be removed. If two communities have the same name and you don't specify the ASCIICOM parameter, the community with ASCIICOM set to *YES is removed.

***YES:** The community name is translated to ASCII characters before it is compared with a community name specified by an SNMP manager.

***NO:** The community name is not translated to ASCII characters before it is compared with a community name specified by an SNMP manager.

Example for RMVCOMSNMP

```
RMVCOMSNMP   COM(ROCHESTER)
```

This command removes community ROCHESTER from the SNMP agent community list.

Error messages for RMVCOMSNMP

***ESCAPE Messages**

TCP4001
Error occurred accessing SNMP configuration information.

TCP4009
Community does not exist.

TCP8050
*IOSYSCFG authority required to use &1.

RMVCFGLE (Remove Configuration List Entries) Command Description

RMVCFGLE Command syntax diagram

Purpose

The Remove Configuration List Entries (RMVCFGLE) command removes entries from a configuration list.

Note: The user may also use the full screen entry display of the Change Configuration List (CHGCFGL) command to add, remove, or change entries in an existing list except for the configuration list TYPE(*SNAPASTHR).

Required Parameters

TYPE Specifies one of the possible configuration list types.

***APPNDIR:** An advanced peer-to-peer networking (APPN) directory search filter configuration list is used. The user can specify one APPN directory search filter entry in the configuration list.

***APPNLCL:** An APPN local location list is used. The user can specify up to 476 APPN local location entries in the configuration list.

***APPNRMT:** An APPN remote location list is used. The user can specify up to 1898 APPN remote location entries in the configuration list.

***APPNSSN:** An APPN session end point filter configuration list is used. The user can specify one APPN session entry in the configuration list.

***ASYNCADR:** An asynchronous network address list is used. The user can specify up to 294 asynchronous network address entries in the configuration list.

***ASYNCLOC:** An asynchronous remote location list is used. The user can specify up to 32000 asynchronous remote location entries in the configuration list.

***RTLPASTHR:** A retail pass-through list is used. The user can key up to 450 retail pass-through entries in the configuration list.

***SNAPASTHR:** An SNA pass-through list is used. The user can key one SNA pass-through entry in the configuration list.

CFGL Specifies the name of the configuration list. This parameter is valid only when *ASYNCADR is specified on the TYPE parameter. Only one of the other configuration list types is allowed on a system. The list types have system-supplied names: QAPPNDIR, QAPPNLCL, QAPPNRMT, QAPPNSSN, QASYNCADR, QASYNCLOC, QRTLPASTHR, QSNAPASTHR.

Optional Parameters

FTRCPNAME
Specifies the control point name of the adjacent control point that is being filtered by the local system when a directory search request is made.

Note: This parameter is valid only if TYPE(*APPNDIR) is specified.

***ANY:** Any control point name is filtered.

generic-filtered-CP-name:* Specify the generic control point name (part of a name followed by an asterisk) of the adjacent control point(s) being filtered. The generic control point name allows one directory entry to be defined for all control points, in a single network, with a name that matches the characters preceding an *.

filtered-CP-name: Specify the control point name of the adjacent control point being filtered.

FTRCPNETID
Specifies the control point network identifier of the adjacent control point being filtered by the local system when a directory search request is made.

Note: This parameter is valid only if TYPE(*APPNDIR) is specified.

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

filtered CP-network-ID: Specify the control point network identifier of the adjacent control point being filtered by the local system.

LCLLOCNAME
Specifies the local location name being supplied by the caller that is being filtered by the local system. When the local system is initiating a session, this is the local location name being used. When a bind is received from another system, this is the Secondary Logical Unit (SLU) name being used.

Note: This parameter is valid only if TYPE(*APPNSSN) is specified.

***ANY:** Any local location name will be filtered by the local system.

generic-local-location-name:* Specify the generic local location name (part of a name followed by an asterisk) of the local location(s) being filtered. The generic local location name allows one entry to be defined for all local location names, on the system, with a name that matches the characters preceding an *.

local-location-name: Specify the local location name that is being filtered by the local system.

APPNLCLE
Specifies the APPN local location entry. If TYPE(*APPNLCL) is specified, this value is required. Up to 50 entries can be specified for this parameter.

APPNRMTE
Specifies the APPN remote location entry. If TYPE(*APPNRMT) is specified, this value is required. Up to 50 entries can be specified for this parameter.

Element 1: Remote Location Name

***ANY:** The system potentially accepts all requests sent to it.

remote-location-name: Specify the remote location of the entry being removed from the configuration list.

Element 2: Remote Network Identifier

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

remote-network-identifier: Specify the remote network identifier of the entry being removed from the configuration list.

Element 3: Local Location Name

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

local-location-name: Specify the local location of the entry being removed from the configuration list.

ASYNCADRE
Specifies the asynchronous network address entry. This value is required if TYPE(*ASYNCADR) is specified. Up to 50 entries can be specified for this parameter.

Note: All entries having the same network address as that specified are removed from the configuration list.

ASYNCLOCE
Specifies the asynchronous remote location entry. This value is required if TYPE(*ASYNCLOC) is specified. Up to 50 entries can be specified for this parameter.

RTLPASTHRE
Specifies the retail pass-through entry. This value is required if TYPE(*RTLPASTHR) is specified. Up to 50 entries can be specified for this parameter.

Note: All entries that have the same retail device name as the one the user specified are removed from the configuration list.

SNAPASTHRE
Specifies the SNA pass-through entry. This value is required if TYPE(*SNAPASTHR) is specified. One group entry can be specified for this parameter.

Example for RMVCFGLE

```
RMVCFGLE  TYPE(*ASYNCLOC)  ASYNCLOCE(RMTLOC1)
```

This command removes the configuration list entry RMTLOC1 from the asynchronous remote location list QASYNCLOC.

Error messages for RMVCFGLE

***ESCAPE Messages**

CPF260F
Configuration list &1 not found.

CPF261C
Index for configuration list &1 not changed.

CPF261D
Index for configuration list &1 not changed.

CPF2625
Not able to allocate object &1.

CPF263A
CFGL type &1 does not match existing type &2.

CPF2634
Not authorized to object &1.

CPF2663
Configuration list &1 previously deleted.

CPF2666
Cannot remove all entries from configuration list &1.

RMVCNNLE (Remove Connection List Entry) Command Description

RMVCNNLE Command syntax diagram

Purpose

The Remove Connection List Entry (RMVCNNLE) command removes an entry from the specified connection list.

Required Parameters

CNNL Specifies the name of the connection list.

ENTRY
Specifies which entry in the connection list is removed.

Example for RMVCNNLE

```
RMVCNNLE   CNNL(CHICAGO)  ENTRY(CORPORATE)
```

This command removes entry CORPORATE from the connection list CHICAGO.

Error messages for RMVCNNLE

***ESCAPE Messages**

CPF266C
Connection list &1 not found.

CPF2685
Entry &2 in connection list &1 not removed.

RMVDEVDMNE (Remove Device Domain Entry) Command Description

RMVDEVDMNE Command syntax diagram

Purpose

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

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

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

Restrictions

1. To use this command you must have *IOSYSCFG authority.
2. This CL command cannot be called from a cluster resource group exit program.
3. The node to be removed cannot be in the recovery domain of any device cluster resource group.
4. The node to be removed and at least one other member of the device domain must be active. On certain conditions, all current members of the device domain must be active.
5. This command will fail if any member of the device domain from which the node being removed has a status of Partition.

Required Parameters

CLUSTER
Specifies the name of the cluster that contains the node.

cluster-name: Specify the name of the cluster which contains the node.

DEVDMN
Specifies the device domain from which the node is being removed.

device-domain-name: Specify the name of the device domain from which the node is being removed.

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

node-identifier: Specify the name of the node.

Example for RMVDEVDMNE

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

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

Error messages for RMVDEVDMNE

<u>*ESCAPE Messages</u>

CPF0001
Error found on &1 command.

«

RD (Remove Directory) Command

RD Command syntax diagram

RD Command	For the description of the RD command, see the RMVDIR (Remove Directory) command description.

RMDIR (Remove Directory) Command

RMDIR Command syntax diagram

RMDIR Command	For the description of the RMDIR command, see the RMVDIR (Remove Directory) command description.

RMVDIR (Remove Directory) Command Description

RMVDIR Command syntax diagram

Purpose

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

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

- RD
- RMDIR

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

Restrictions:

1. To remove a directory, the user must have use and object existence authority for the specified directory and object existence authority for every object in it. The user must also have execute authority to the directory that contains the directory to be removed. If you do not have object existence authority for the directory being removed, nothing is removed or unlinked. If you do not have object existence authority for one or more objects in the directory, those objects are not unlinked and the directory is not removed.
2. You cannot remove a directory if it is the current directory for your job.
3. This command cannot be used to delete reserved directories and libraries.
4. When an object is in use in QSYS.LIB, » independent ASP QSYS.LIB « or QDLS, the object cannot be unlinked. When an object is in use in QOpenSys or the root file system, the object is successfully unlinked, and the object is deleted when no longer in use.

Required Parameter

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

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

Optional Parameter

RMVLNK

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

***NO:** Only an empty directory is removed. A directory may contain entries for the directory (.) and for the parent directory (..) and still be treated as an empty directory.

***YES:** All objects within the specified directory are deleted. If the file system that contains the directory does not support removal of links in the directory, the error message CPFA0AC “Request cannot be completed” is sent.

Example for RMVDIR

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

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

Error messages for RMVDIR

***ESCAPE Messages**

CPFA085

Home directory not found for user &1.

CPFA093

Name matching pattern not found.

CPFA09C

Not authorized to object.

CPFA09D

Error occurred in program &1.

CPFA09E

Object in use.

CPFA0A1

An input or output error occurred.

CPFA0A3

Path name resolution causes looping.

CPFA0A7

Path name too long.

CPFA0A9

Object not found.

CPFA0AB

Object name not a directory.

CPFA0AC

Request cannot be completed. Directory contains objects.

CPFA0AD
Function not supported by file system.

CPFA0B1
Requested operation not allowed. Access problem.

CPFA0B2
No objects satisfy request.

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

RMVDIRE (Remove Directory Entry) Command Description

RMVDIRE Command syntax diagram

Purpose

The Remove Directory Entry (RMVDIRE) command removes a specific entry from the system distribution directory. When a user ID and address is removed from the directory, it is also removed from all distribution lists and the list of nicknames for that user. However, the user ID and address for that user is not removed in the list of nicknames for other users.

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

If a user ID and address has multiple descriptions associated with it, options exist to remove only a specific description or all descriptions.

The RMVDIRE command does not provide interactive display support. This is provided by the Work with Directory Entries (WRKDIRE) command.

If the specified user ID, address, and description do not exist in the directory, an error message is returned.

Restrictions:

1. The person running this command must have security administrator (*SECADM) authority.
2. The user ID and address being removed must not:
   - Be enrolled in OfficeVision
   - Have ownership of documents or folders in the Document Interchange Architecture (DIA) library

   If either of the above conditions are not met, the user ID and address is not removed from the directory.
3. If the user ID and address being removed has not received mail from the mail queue, the following can happen:
   - If the request is entered interactively, a status message is sent asking whether the queue should be cleared. If the user responds yes, the mail is cleared from the queue and the user is removed from the directory. If the user responds no, the user ID and address is not removed from the directory.
   - If the request is not entered interactively (for example, the request is from a batch CL program), the user ID and address is not removed from the directory.

Required Parameter

USRID
Specifies the user ID and address for the user entry to be removed.

Element 1: User ID

user-ID: Specify the user ID that is removed. Up to 8 characters can be specified.

Element 2: User Address

user-address: Specify the user address to be removed. Up to 8 characters can be specified.

Optional Parameters

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

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

***ALL:** All entries with the specified user ID and address are removed.

user-description: Specify up to 50 characters for the description for the user. This description must be the same as the one in the directory for this user ID and address.

CMDCHRID

Specifies the character identifier (graphic character set and code page) for data being specified as parameter values on this command. This character identifier (CHRID) is related to the display device used to specify the command. More information about CHRID processing is in the Application Display Programming book.

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

***DEVD:** The system determines the graphic character set and code page values for the command parameter from the display device description where the command is entered. This option is valid only when specified from an interactive job. If this value is specified in an interactive CL program or a batch job, an error message is sent.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values used to create the command parameters. Valid values range from 1 through 9999 characters.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Example for RMVDIRE

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

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

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

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

Error messages for RMVDIRE

*ESCAPE Messages

CPF89A3

Operation not successful due to authority reasons.

CPF90A8
Operation not successful due to data validation reasons.

CPF90A8
*SECADM special authority required to do requested operation.

CPF9005
System resource required to complete this request not available.

CPF9009
System requires file &1 in &2 be journaled.

CPF9024
System cannot get correct record to finish operation.

CPF905C
Error occurred trying to find a translation table.

CPF9087
Directory entry not removed.

CPF9096
Cannot use CMDCHRID(*DEVD), DOCCHRID(*DEVD) in batch job.

CPF9810
Library &1 not found.

CPF9838
User profile storage limit exceeded.

CPF9845
Error occurred while opening file &1.

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

RMVDIRSHD (Remove Directory Shadow System) Command Description

RMVDIRSHD Command syntax diagram

Purpose

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

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

Required Parameter

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

Optional Parameter

RMVDTA

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

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

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

Example for RMVDIRSHD

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

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

Error messages for RMVDIRSHD

***ESCAPE Messages**

CPF89A5

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

CPF90A8

*SECADM special authority required to do requested operation.

CPF905C

Error occurred trying to find a translation table.

CPF9838

User profile storage limit exceeded.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

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

»

RMVDSTCLGE (Remove Distribution Catalog Entry) Command Description

Note: To use this command, you must have the 5722-MG1 (Managed System Services for iSeries) licensed program installed.

RMVDSTCLGE Command syntax diagram

Purpose

The Remove Distribution Catalog Entry (RMVDSTCLGE) command allows the user to remove a specific catalog entry from the distribution catalog. If the associated data object is stored in the distribution repository, the repository object is also deleted.

Required Parameter

GLBNAME

Specifies the tokens of the global name used to find the unique catalog entry to be removed.

Element 1: Token 1

***NETID:** The first global name token value is a network ID generated by the command from the network attributes.

***SERVER:** This token is stored within the change request activity with the value &SERVER, and replaced by the short name of the change control server when the object is distributed.

***TARGET:** This token is stored within the change request activity with the value &TARGET, and replaced by the short name of the target when the object is distributed.

***MDDATE:** This token is stored within the change request activity with the value &DATE, and replaced when distributed by the date the object was last modified.

***MDTIME:** This token is stored within the change request activity with the value &TIME, and replaced when distributed by the time the object was last modified.

global-name-token-1: Specify the first token of the global name.

Element 2-10: Token 2-10

***ANY:** The corresponding token value is ignored.

***HIGHEST:** The catalog entry with the highest corresponding token value is selected.

***LOWEST:** The catalog entry with the lowest corresponding token value is selected.

***CPNAME:** The global name token value is a control point name. This value is generated from the network attributes.

***NETID:** The global name token *n* value is a network ID. This value is generated from the network attributes.

***SERVER:** This token is stored within the change request activity with the value &SERVER, and replaced by the short name of the change control server when the object is distributed.

***TARGET:** This token is stored within the change request activity with the value &TARGET, and replaced by the short name of the target when the object is distributed.

***MDDATE:** This token is stored within the change request activity with the value &DATE, and replaced when distributed by the date the object was last modified.

***MDTIME:** This token is stored within the change request activity with the value &TIME, and replaced when distributed by the time the object was last modified.

global-name-token-n: Specify a token of the global name.

Example for RMVDSTCLGE

```
RMVDSTCLGE  GLBNAME(ABCCO *LOWEST *HIGHEST *ANY XYZ)
```

This command removes an entry from the catalog. Tokens one and five match the corresponding catalog entry tokens. The catalog entry with the lowest value is selected for the second token. The highest value is selected for the third token. Token 4 is ignored. The priority of the search examines the tokens from left to right.

```
RMVDSTCLGE  GLBNAME(CMPNY1 *SERVER *TARGET *MDDATE *MDTIME)
```

This command removes the entry with global name CMPNY1 &SERVER &TARGET &DATE &TIME

Error messages for RMVDSTCLGE

*ESCAPE Messages

MSS0066
Internal processing error occurred.

MSS0111
Distribution repository object not available.

MSS0112
Distribution repository not available.

MSS0113
Not authorized to distribution repository.

MSS0116
Maximum global name length exceeded.

MSS0117
Global name token &3 not valid. Reason code &4.

MSS011B
Distribution catalog entry not found.

MSS011C
Distribution catalog not available.

MSS011D
More than one distribution catalog entry found.

MSS0121
Distribution repository not found.

MSS0122
Distribution repository damaged.

MSS0123
Internal processing error occurred.

MSS0124
Error while managing distribution catalog.

MSS0125
Distribution catalog damaged.

MSS0173
Not authorized to delete repository object.

MSS0174
Distribution catalog entry not removed.

MSS0176
Distribution catalog entry not removed.

MSS0904
Number of tokens in global name not valid.

MSS0905
Token &2 not valid for global name.

MSS0906
Global name contains more than one date or time tokens.

«

RMVDSTLE (Remove Distribution List Entry) Command Description

RMVDSTLE Command syntax diagram

Purpose

The Remove Distribution List Entry (RMVDSTLE) command removes entries from an existing distribution list. Up to 300 entries can be removed from a list at one time.

Restriction: The user must have security administrator authority (*SECADM) to remove entries from a distribution list owned by another user. Users can remove entries from distribution lists that they created.

Required Parameters

LSTID Specifies the two-part list identifier of the distribution list that is to have entries removed.

Element 1: List Identifier

list-ID: Specify the list identifier (ID) of the distribution list.

Element 2: List Qualifier

list-ID-qualifier: Specify the list ID qualifier of the distribution list.

Note: The distribution list identifier has two parts, the ID and the qualifier, separated by at least one space. If lowercase characters are specified, the system changes them to uppercase.

The naming rules for the two-part list ID are identical to the rules for the user ID and address. A complete description of these rules is in the SNA Distribution Services book.

USRID

Specifies the user ID, address, and description of the users for whom distribution list entries are removed. Both the user ID and address must be provided and must be separated by at least one space. If any lowercase characters are specified, the system translates and stores them as uppercase. The description can be entered to specify the removal of a specific user ID.

A list ID and address can be used in place of the user ID and address to identify a remote distribution list that is removed from the distribution list.

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

Element 1: User ID

user-ID: Specify the user ID (or the valid list ID for a remote list entry) of the user for whom the entry is removed.

Element 2: User Address

user-address: Specify the user address (or the valid list address for a remote list entry) of the user for whom the entry is removed.

Element 3: User Description

***FIRST:** The first description in the specified user ID and address (or list ID and address) is removed. If only one entry exists, it is the one removed from the list.

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

user-description: Specify the description for the user. If a list ID is specified, specify the list description. The description can be up to 50 characters in length.

Optional Parameter

CMDCHRID

Specifies the character identifier (graphic character set and code page) for data being specified as parameter values on this command. This character identifier (CHRID) is related to the display device used to specify the command. More information about CHRID processing is in the Application Display Programming book.

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

***DEVD:** The system determines the graphic character set and code page values for the command parameter from the display device description where the command is entered. This option is valid only when specified from an interactive job. If this value is specified in an interactive CL program or a batch job, an error message is sent.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values used to create the command parameters. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Example for RMVDSTLE

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

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

Additional Considerations

To remove entries from a list ID, the list ID must exist in the directory. If the list does not exist, an error message is returned.

Up to 300 sets of user IDs, addresses and user descriptions can be entered on this command. Each set of user IDs and addresses is examined to determine whether it exists in the directory. If no such entry exists, an error message is returned. If an entry is found, it is removed from the list.

Error messages for RMVDSTLE

***ESCAPE Messages**

CPF9024

System cannot get correct record to finish operation.

CPF905C

Error occurred trying to find a translation table.

CPF9092

No entries removed from list &1 &2.

CPF9093
&1 entries removed from list &2 &3.

CPF9096
Cannot use CMDCHRID(*DEVD), DOCCHRID(*DEVD) in batch job.

CPF9838
User profile storage limit exceeded.

CPF9845
Error occurred while opening file &1.

RMVDSTQ (Remove Distribution Queue) Command Description

RMVDSTQ Command syntax diagram

Purpose

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

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

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:

1. This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
2. The following distribution queues cannot be removed:
   - Queues referred to in the routing table
   - Queues that contain distributions waiting to be sent
   - DLS (document library services) queues that have remote libraries configured to use them
   - SVDS (SystemView distribution services) queues when a receiver is active or when distributions have been received and the sender has not acknowledged receiving confirmation.
3. Messages that report errors about distribution queues may display or print different characters than the user 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 on the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the 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).

Required Parameter

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

Example for RMVDSTQ

```
RMVDSTQ   DSTQ(CHICAGO)
```

This command removes the distribution queue entry named CHICAGO.

Error messages for RMVDSTQ

***ESCAPE Messages**

CPF8802
Distribution queue &1 was not found.

CPF8807
Error occurred while using QSNADS journal.

CPF8809
Errors detected on SNADS internal queues.

CPF881B
Distributions are being received for distribution queue &1

CPF8826
Distribution queue entries exist for distribution queue &1.

CPF8827
Routing table entries exist for distribution queue &1.

CPF8828
Remote document library entries exist for *DLS distribution queue &1.

CPF8833
Distribution queue &1 already exists.

CPF8849
Queue &1 in use by another distribution services function.

CPF9845
Error occurred while opening file &1.

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

CPF9847
Error occurred while closing file &1 in library &2.

CPF9899
Error occurred during processing of command.

RMVDSTRTE (Remove Distribution Route) Command Description

RMVDSTRTE Command syntax diagram

Purpose

The Remove Distribution Route (RMVDSTRTE) command removes a remote system from the distribution services routing table. Once a system is removed from the table, distributions can no longer be sent to that system from the local system.

Interactive display support is provided by the Configure Distribution Services (CFGDSTSRV) command.

More information on configuring a distribution network is in the SNA Distribution Services book.

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

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

Required Parameter

SYSNAME

Specifies the system name and group name of the remote system being removed from the routing table.

Element 1: System Name

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

system-name: Specify a maximum of 8 characters for the name of the remote system.

Element 2: System Group Name

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

system-group-name: Specify a maximum of 8 characters for the system group name. The system name and group name must be separated by at least one blank.

Examples for RMVDSTRTE

Example 1: Removing a System from the Routing Table

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

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

Example 2: Removing a Generic Routing Table Entry

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

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

Error messages for RMVDSTRTE

*ESCAPE Messages

CPF8802

Distribution queue &1 was not found.

CPF8807

Error occurred while using QSNADS journal.

CPF8815

Routing table entry &1 &2 not found.

CPF8831

Entry &1 &2 already exists in routing table.

CPF8837

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

CPF8849

Queue &1 in use by another distribution services function.

CPF9845

Error occurred while opening file &1.

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

CPF9847
Error occurred while closing file &1 in library &2.

CPF9899
Error occurred during processing of command.

RMVDSTSYSN (Remove Distribution Secondary System Name) Command Description

RMVDSTSYSN Command syntax diagram

Purpose

The Remove Distribution Secondary System Name (RMVDSTSYSN) command removes an entry from the distribution services secondary system name table. The table contains the names of all of the alternate (or alias) systems for which the local system receives distributions. When an alternate system name is removed from the table, the local system no longer receives distributions for the alternate system.

Interactive display support is provided by the Configure Distribution Services (CFGDSTSRV) command.

More information about configuring a distribution network is in the SNA Distribution Services book.

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

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

Required Parameter

SYSNAME
Specifies the alternate system name and system group name being removed from the distribution services secondary system name table.

Element 1: System Name

system-name: Specify the name of the system for which this system is no longer to receive distributions.

Element 2: System Group Name

system-group: Specify the system group name of the system for which this system is no longer to receive distributions. The system name and group name must be separated by at least one blank.

Example for RMVDSTSYSN

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

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

Error messages for RMVDSTSYSN

***ESCAPE Messages**

CPF8807
Error occurred while using QSNADS journal.

CPF8818
Secondary system name table entry &1 &2 not found.

CPF8835
System name/Group &1 &2 already specified.

CPF8837
System name/Group &1 &2 in use by another distribution services function.

CPF9845
Error occurred while opening file &1.

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

CPF9847
Error occurred while closing file &1 in library &2.

CPF9899
Error occurred during processing of command.

RMVDLOAUT (Remove Document Library Object Authority) Command Description

RMVDLOAUT Commnd syntax diagram

Purpose

The Remove Document Library Object Authority (RMVDLOAUT) command is used to remove an existing user's authority to documents or folders.

The following types of authority can be removed:

- An existing specific user's authority
- An existing authorization list's authority to the associated document library object
- An existing object access code associated with the document library object

Restriction: The user of this command must have *ALL authority to the objects, be the owner of the objects, or have *ALLOBJ authority.

Required Parameter

DLO Specifies the document or folder from which authority is removed.

***SYSOBJNAM:** The system object name specified in the SYSOBJNAM parameter has user authority removed.

***ALL:** All objects in a specified folder have user authority removed. If DLO(*ALL) is specified, then the FLR parameter must be specified.

document-library-object-name: Specify the user-assigned name of the document or folder from which user authority is removed. Up to 12 characters can be specified.

Optional Parameters

FLR Specifies the name of the folder that contains the document.

***NONE:** A folder name is not specified. If DLO(name) is specified and the object is located in a folder, then FLR(*NONE) cannot be specified. If DLO(*ALL) is specified, then FLR(*NONE) cannot be specified.

folder-name: Specify the user-assigned name of the folder in which the object specified in the DLO parameter is located. The name can consist of a series of folder names if the folder containing the object is located in another folder. Up to 63 characters can be specified.

USER Specifies the name of a user whose specific authority is removed.

***SAME:** The value does not change.

***ALL:** Specific user authority for all users is removed.

user-profile-name: Specify the name of the user profile from which specific authority is removed. Up to 50 user profile names can be specified.

AUTL Specifies that the authority specified in the existing authorization list for the object named in the OBJ parameter is removed.

***SAME:** The value does not change.

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

ACC Specifies the access codes to be removed for the object.

***SAME:** The value does not change.

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

access-code: Specify which currently assigned access codes (ranging from 0 through 2047) are removed. Up to 50 access codes can be specified. Access code zero (0) stands for public access of *USE.

SYSOBJNAM
Specifies the system object name. This parameter is valid only when DLO(*SYSOBJNAM) or DOCL(*SYSOBJNAM) is specified. A full ten characters must be specified.

Example for RMVDLOAUT

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

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

Error messages for RMVDLOAUT

***ESCAPE Messages**

CPF8A75
Not authorized to access folder &1.

CPF8A77
Folder &1 not found.

CPF8A78
Folder &1 in use.

CPF8A79
Folder &1 is logically damaged.

CPF8A80
Document &2 in use in folder &1.

CPF8A82
Document &2 not found in folder &1.

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

CPF8A88
Operation not allowed on document &2 in folder &1.

CPF8A89
Document &2 in folder &1 is logically damaged.

CPF90BA
Authority request for document library object failed.

CPF9073
No authority to view or change the security of document library object &1.

CPF908A
Requester &1 not enrolled.

CPF908B
Document library object not found.

CPF908E
&1 objects changed; &2 objects not changed.

CPF909A
Document &2 in folder &1 is damaged.

CPF9095
Folder &1 is damaged.

RMVEMLCFGE (Remove Emulation Configuration Entry) Command Description

RMVEMLCFGE Command syntax diagram

Purpose

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

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

Required Parameter

EMLCFGE
Specifies the name of the configuration entry you are removing.

Example for RMVEMLCFGE

```
RMVEMLCFGE   EMLCFGE(FASBPRINT)
```

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

Error messages for RMVEMLCFGE

***ESCAPE Messages**

CPF85FA
Emulation entry QEMDFTCFGE was not removed.

CPF853E
Emulation configuration entry &1 not found.

CPF854B
Internal error in emulation configuration routines.

RMVENVVAR (Remove Environment Variable) Command Description

RMVENVVAR Command syntax diagram

Purpose

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

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

Required Parameter

ENVVAR
Specifies the name of the environment variable to be removed. If an environment variable by this name does not exist at the specified level (LEVEL parameter), error message CPFA981 is issued. If the special value of *ALL is specified, then all environment variables at the specified level (LEVEL parameter) are removed. The case is preserved when lowercase characters are specified. If an apostrophe is intended, two apostrophes must be used ('')

Optional Parameter

LEVEL
Specifies the level of the environment variable.

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

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

Examples for RMVENVVAR

Example 1: Remove a Job-level Environment Variable

```
RMVENVVAR   ENVVAR(altdir)
```

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

Example 2: Remove a System-level Environment Variable

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

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

Example 3: Remove All Job-level Environment Variables

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

This commands removes all job-level environment variables.

Example 4: Remove All System-level Environment Variables

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

This commands removes all system-level environment variables.

Error messages for RMVENVVAR

***ESCAPE Messages**

CPFA981
Environment variable does not exist.

CPFA98E
*JOBCTL special authority required to update system-level environment variables.

RMVEXITPGM (Remove Exit Program) Command Description

RMVEXITPGM Command syntax diagram

Purpose

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

Required Parameters

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

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

PGMNBR
Specifies the program number of the exit program being removed.

***ALL:** All exit programs for the specified exit point format (FORMAT parameter) from the specified exit point (EXITPNT parameter) are removed.

program-number: Specify the program number of the exit program being removed.

Example for RMVEXITPGM

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

This command removes the exit program for exit point USER_EXIT_ONE that was added with program sequence number 1 for exit point format EXIT1.

No error messages.

RMVEWCBCDE (Remove Extended Wireless Controller Bar Code Entry) Command Description

RMVEWCBCDE Command syntax diagram

Purpose

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

Restriction: You must have *IOSYSCFG special authority to use this command.

Required Parameters

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

INZMBR

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

Optional Parameter

INZFILE

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

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

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

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

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

QEWCSRC: The source file name QEWCSRC is used.

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

Example for RMVEWCBCDE

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

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

No error messages.

RMVEWCPTCE (Remove Extended Wireless Controller PTC Entry) Command Description

RMVEWCPTCE Command syntax diagram

Purpose

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

Restriction: You must have *IOSYSCFG special authority to use this command.

Required Parameters

PTCGRP

Specifies the PTC group name of the entry being removed.

INZMBR

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

Optional Parameter

INZFILE

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

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

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

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

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

QEWCSRC: The source file name QEWCSRC is used.

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

Example for RMVEWCPTCE

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

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

No error messages.

RMVFTRACNE (Remove Filter Action Entry) Command Description

RMVFTRACNE Command syntax diagram

Purpose

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

Required Parameters

FILTER

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

The name of the filter 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.

filter-name: Specify the name of the filter.

GROUP
Specifies the group that identifies the action entry being removed.

Example for RMVFTRACNE

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

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

Error messages for RMVFTRACNE

***ESCAPE Messages**

CPF2150
Object information function failed.

CPF2151
Operation failed for &2 in &1 type *&3.

CPF812F
Filter damaged.

CPF91DD
Action entry for group &4 not found.

CPF91EC
Internal processing error occurred.

CPF91E8
Internal processing error occurred.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9830
Cannot assign library &1.

RMVFTRSLTE (Remove Filter Selection Entry) Command Description

RMVFTRSLTE Command syntax diagram

Purpose

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

Required Parameters

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

The name of the filter 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.

filter-name: Specify the name of the filter that is used.

SEQNBR

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

Example for RMVFTRSLTE

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

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

Error messages for RMVFTRSLTE

***ESCAPE Messages**

CPF2150
Object information function failed.

CPF2151
Operation failed for &2 in &1 type *&3.

CPF812F
Filter damaged.

CPF91DC
Selection entry with sequence number &4 not found.

CPF91EC
Internal processing error occurred.

CPF91E8
Internal processing error occurred.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9830
Cannot assign library &1.

RMVFNTTBLE (Remove Font Table Entry) Command Description

RMVFNTTBLE Command syntax diagram

Purpose

The Remove Font Table Entry (RMVFNTTBLE) command removes an entry in the specified font table. This command removes an entry in the user font mapping tables used by PSF/400 that controls:

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

The entry must have first been added to the user tables with the ADDFNTTBLE (Add Font Table Entry) command.

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

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

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

Refer to Appendix D in the Printer Device Programming book for more information on font mapping tables.

Restriction: The PSF/400 feature is required to use this command.

Required Parameter

FNTTBL

Specifies the name of the font table to be changed (removes an entry).

***PHFCS:** The printer resident to host resident font character set table is to be changed (removes an entry).

This table would be used when your application (like Office Vision/400, DDS, etc) references printer resident fonts and the printer (3827, 3825, 3820, 3900 Model 1, etc) does not support resident fonts. PSF/400 must map the references from printer resident fonts to host resident fonts and down load them.

***PHCP:** The printer resident to host resident code page mapping table is to be changed (removes an entry).

This table is like the QPHFCS table, in that it is used when the application references printer resident code pages and the printer being used does not support printer resident code pages. The printer resident code page must be mapped to a host resident code page and down loaded to the printer by PSF/400.

***HPFCS:** The host resident to printer resident font character set table is to be changed (removes an entry).

This table is used when your application references host resident fonts (font character sets and code pages) and the printer (4224, 4234, 4230, 64XX,etc) does not support down loading of host resident fonts. PSF/400 must map the references from host resident fonts to printer resident fonts.

***HPCP:** The host resident to printer resident code page mapping table is to be changed (removes an entry).

This table is like the QHPFCS table, in that it is used when the application references host resident code pages and the printer being used does not support host resident code pages. The host resident code page must be mapped to a printer resident code page and down loaded to the printer by PSF/400.

The name of the font table must be specified when a printer resident to printer resident font substitution table is to be changed. This printer resident font substitution table should be used when all three of the following conditions exit.

- You are printing to a PSF/400 attached printer
- Your application specifies a printer resident font which is not supported by the printer you are using.
- You want to specify a different substitute printer resident font than the one selected by the system.

To use a printer resident to printer resident font substitution table wit a particular PSF/400 printer, you need to specify the name of the font table on the FNTTBL parameter of the Create PSF Configuration (CRTPSFCFG) or Change PSF Configuration (CHGPSFCFG) command.

The name of the printer resident to printer resident font substitution table can be qualified by one of the following library values:

***CURLIB:** The current library is used to store the font table. 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 you want to store the font table.

font-table-name: Specify the name of the printer resident to printer resident font substitution table to be created.

Optional Parameters

PHFCS

Specifies the printer resident font entry to be removed.

Element 1: Font Identifier

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

Element 2: Font Width

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

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

font-width: Specify a width for the font identifier. When mapping a fixed pitch raster font (1 - 750, 3840 - 4095), a width should be specified. The point size value can be *WIDTH or a value can be given. See Appendix D in the Printer Device Programming book for more information on font widths for printer resident fonts.

Element 3: Font Attributes

***NONE:** No special font attributes are specified on this font.

***BOLD:** The printer resident font is a bold font.

***ITALIC:** The printer resident font is an italic font.

***BOLDITC:** The printer resident font is a bold italic font.

***DBLWIDE:** The printer resident font is a double wide font.

***ITCDBLWIDE:** The printer resident font is an italic double wide font.

Element 4: Graphic Character Set

Specifies the graphic character set for the font identifier. This parameter allows you to map a font identifier to more than one host font character set (based upon graphic character set). The value specified is mapped to a superset graphic character set. See the Printer Device Programming book for information on superset graphic character sets.

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

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

Element 5: Point Size

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

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

point-size: Specify a point size ranging from 0.1 through 999.9. When mapping a typographic raster font (2304 - 3839, 4096 - 53247, 61440 - 65534), a point size value should be specified.

HPFCS

Specifies the host resident font entry to be removed.

Element 1: Host Font Character Set

font-character-set: Specify the font character set.

Element 2: Font Type

***RASTER:** The host resident font is a raster font.

***OUTLINE:** The host resident font is a outline font.

PPFCS

Specifies the printer resident to printer resident font substitute entry to be removed.

Element 1: Font Identifier

font-identifier: Specify the printer resident font identifier to be removed.

Element 2: Point Size

***NONE:** No font point size is specified. This should be specified for all non-scalable fonts.

***ALL:** Specifies that all point sizes for a scalable font will be mapped. If the font is not scalable, then this will treated the same as *NONE.

point-size: Specify a point size ranging from 0.1 through 999.9.

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

Element 1: Graphic Character Set

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

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

Element 2: Code Page

code-page: Specify the printer resident code page value

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

Element 1: Host Code Page

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

Examples for RMVFNTTBLE

Example 1: Remove Font Entry

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

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

Example 2: Remove Code Page Entry

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

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

Error messages for RMVFNTTBLE

***ESCAPE Messages**

CPF2182

Not authorized to library &1.

CPF2283

Authorization list &1 does not exist.

CPF88D2
Font table &1 in library &2 not changed.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

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

RMVHSTJS (Remove History using Job Scheduler) Command Description

Note: To use this command, you must have the 5722-JS1 (Job Scheduler for iSeries) licensed program installed.

RMVHSTJS Command syntax diagram

Purpose

The Remove History using Job Scheduler (RMVHSTJS) command allows you to remove history from the Job Scheduler job history file by specifying the amount of history you want to keep or remove. You can specify whether you want to keep a number of job history occurrences or you can remove history entries based on a date range that you specify.

Optional Parameters

RMVTYPE
Specifies how entries are to be selected from the Job Scheduler history log.

***OCCUR:** History file entries are kept based on the number of occurrences specified in the NBROCCUR parameter.

***DAYS:** History file entries are kept based on a date range specified in the SLTDATE parameter.

NBROCCUR
Specifies how many job history occurrences are to be kept in the history file. The number of occurrences begins with the oldest (earliest date) occurrence and progresses to the most recent. Occurrences can range from 1 to 999.

180: The number of occurrences for a job to be kept in the file is 180.

number-of-occurrences: Specify the number of occurrences for a job that are to be kept in the file.

SLTDATE
Specifies the range of dates that you want to remove from Job Scheduler history information. The **From date** is the beginning point in the Job Scheduler history file and the **To date** is the ending point.

Element 1: From date

***BEGIN:** Uses the earliest Job Scheduler history entry as the beginning point into the Job Scheduler history.

***CURRENT:** Uses the current date as the beginning entry point into the Job Scheduler history.

from-date: Specify the date in job date format with or without date separators, that is, 12/01/99 or 120194, that you want to be the beginning date of the range of dates.

number-of-days: Specify a number to represent the number of days before the current day that you want to use for the search. An example would be 10. If today's date were 3/12/99 (date format is mm/dd/yy), the From date for the Job Scheduler history file would be 3/2/99.

Element 2: To date

180: The number of days of history that remains is 180.

***CURRENT:** Uses the current date as the ending date of the date range.

***END:** The last date in the history file is used as the ending date for the date range.

to-date: Specify the date with or without date separators, that is, 12/01/99 or 120199, that you want to use as the ending date of the date range.

number-of-days: Specify a number to represent the number of days before the current day that you want to use for the search. An example would be 10. If today's date were 3/22/99 (date format is mm/dd/yy), the To date for the Job Scheduler history file would be 3/12/99.

Example for RMVHSTJS

Example 1: Removing All but 60 Days of History Information

```
RMVHSTJS RMVTYPE(*DAYS) SLTDATE(*BEGIN 60)
```

In this example, 60 days of history information is kept. Information from the beginning of the history information that is older than 60 days is removed.

Error messages for RMVHSTJS

None »

RMVIMGCLGE (Remove Image Catalog Entry) Command Description

RMVIMGCLGE Command syntax diagram

Purpose

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

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

Restriction:

1. You must have *SECADM and *ALLOBJ special authorities to use this command.

Required Parameters

IMGCLG

Specifies the image catalog name that the image is to be removed from.

image-catalog-name: Specify the catalog name that the image is to be removed from.

IMGCLGIDX

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

image-catalog-index-number: Specify an index number from 1 to 64.

Optional Parameter

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

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

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

Examples for RMVIMGCLGE

Example 1: Removing an image catalog entry

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

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

Example 2: Remove an image catalog entry and delete the image file

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

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

Error messages for RMVIMGCLGE

***ESCAPE Messages**

CPFBC30
Catalog entry not removed from image catalog &1.

CPFBC40
Not authorized to command &1.

CPFBC41
&1 command failed.

«

RMVICFDEVE (Remove Intersystem Communications Function Program Device Entry) Command Description

RMVICFDEVE Command syntax diagram

Purpose

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

Required Parameters

FILE Specifies the qualified name of the ICF file from which the device entries are removed.

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

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

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

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

file-name: Specify the name of the icf file from which the device entries are removed.

PGMDEV
Specifies the names of up to 50 program devices that are removed from the ICF file.

Example for RMVICFDEVE

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

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

Error messages for RMVICFDEVE

***ESCAPE Messages**

CPF7366
Devices not removed from file &1 in &2.

RMVIPSIFC (Remove IP over SNA interface) Command Description

RMVIPSIFC Command syntax diagram

Purpose

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

Restrictions:

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

Required Parameter

INTNETADR
Specifies the internet address of an interface that had previously been added to the SNA configuration with the Add IP over SNA Interface (ADDIPSIFC) CL command. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

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

Example for RMVIPSIFC

```
RMVIPSIFC   '9.5.1.248'
```

This command removes the interface with the IP address 9.5.1.248.

Error messages for RMVIPSIFC

***ESCAPE Messages**

TCP8050
*IOSYSCFG authority required to use &1.

TCP9999
Internal system error in program &1.

RMVIPSLOC (Remove IP over SNA Location Entry) Command Description

RMVIPSLOC Command syntax diagram

Purpose

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

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

Required Parameters

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

SUBNETMASK
Specifies the subnet mask of the location entry to be removed.

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

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

Examples for RMVIPSLOC

Example 1: Removing a Location Entry for a Subnetwork

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

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

Example 2: Removing a Location Entry for a Network

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

This command removes a location entry for network 128.3.

No error messages.

RMVIPSRTE (Remove IP over SNA Route) Command Description

RMVIPSRTE Command syntax diagram

Purpose

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

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

Required Parameters

RTEDEST

Specifies the route destination of the route to be removed. The route destination is specified in the form *nnn.nnn.nnn.nnn* where *nnn* is a decimal number ranging from 0 to 255.

If the route destination address is entered from a command line, the address must be enclosed in apostrophes.

SUBNETMASK

Specifies the subnet mask of the route to be removed.

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

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

NEXTHOP

Specifies the next hop of the route to be removed. The next hop is specified in the form *nnn.nnn.nnn.nnn* where *nnn* is a decimal number ranging from 0 to 255.

If the next hop address is entered from a command line, the address must be enclosed in apostrophes.

Example for RMVIPSRTE

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

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

Error messages for RMVIPSRTE

***ESCAPE Messages**

TCP264A

&2 &1 not removed.

TCP8050

*IOSYSCFG authority required to use &1.

TCP9999

Internal system error in program &1.

RMVJOBQE (Remove Job Queue Entry) Command Description

RMVJOBQE Command syntax diagram

Purpose

The Remove Job Queue Entry (RMVJOBQE) command removes a job queue entry from the specified subsystem description. This command can be used to remove the current job queue entry so that a different job queue can be assigned. Jobs on the queue remain on the queue for processing when the queue is reassigned to a subsystem description and the subsystem is started.

Restrictions:

1. Users of this command must have object operational and object management authorities for the specified subsystem description.
2. A job queue entry cannot be removed if any currently active jobs were started from the job queue.

Required Parameters

SBSD Specifies the qualified name of the subsystem description from which the job queue entry is removed.

The name of the subsystem description 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.

subsystem-description-name: Specify the name of the subsystem description.

JOBQ Specifies the qualified name of the job queue whose job queue entry is removed from the subsystem description.

The name of the job queue 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.

job-queue-name: Specify the name of the job queue.

Example for RMVJOBQE

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

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

Error messages for RMVJOBQE

***ESCAPE Messages**

CPF1619
Subsystem description &1 in library &2 damaged.

CPF1691
Active subsystem description may or may not have changed.

CPF1697
Subsystem description &1 not changed.

RMVJOBSCDE (Remove Job Schedule Entry) Command Description

RMVJOBSCDE Command syntax diagram

Purpose

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

Restriction: To remove entries, you must have *JOBCTL special authority; otherwise you can remove only those entries that you added.

Required Parameter

JOB Specifies the name of the job schedule entry.

job-name: Specify the name of the job schedule entry.

generic-job-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. For more information on the use of generic names, refer to generic names. If a generic name is specified, ENTRYNBR(*ALL) must also be specified.

Optional Parameter

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

***ONLY:** One entry in the job schedule has the job name specified on the JOB parameter. If *ONLY is specified and more than one entry has the specified job name, no entries are removed and a message is sent.

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

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

Examples for RMVJOBSCDE

Example 1: Removing Job Schedule Entries

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

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

Example 2: Removing an Individual Job Schedule Entry

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

This command removes the job PAYROLL in the job schedule.

Example 3: Removing a Generic Job Schedule Entry

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

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

Error messages for RMVJOBSCDE

***ESCAPE Messages**

CPF1628
Job schedule entry &3 number &4 not found.

CPF1629
Not authorized to job schedule &1.

CPF1630
Not authorized to job schedule entry &3 number &4.

CPF1631
&3 entries successfully removed, &4 entries not removed.

CPF1636
More than one entry with specified entry job name found.

CPF1637
Job schedule &1 in library &2 in use.

CPF1638
Job schedule entry &3 number &4 in use.

CPF1640
Job schedule &1 in library &2 does not exist.

CPF1641
Job schedule &1 in library &2 damaged.

CPF1645
No job schedule entries found for specified name.

CPF1646
Entry number must be *ALL when generic name specified.

RMVJOBJS (Remove Job using Job Scheduler) Command Description

RMVJOBJS Command syntax diagram

Purpose

The Remove Job using Job Scheduler (RMVJOBJS) command allows you to remove a job from the job schedule.

Note: When referring to a job in this command, we are referring to an entry in Job Scheduler.

An **entry** in Job Scheduler is a user-defined name for commands or programs that you want to process at scheduled times and dates. Job Scheduler jobs (entries) are not OS/400 objects.

The job can belong to a job group.

Note: You must remove all subordinate job groups before a job with sequence number 1 can be removed.

Required Parameter

JOB Specifies the name of the job schedule entry that you want to remove. You must specify a job and optionally can specify a group to which the job belongs as well as a sequence number for the job within the group.

Element 1: Job

job-name: Specify the name of the job that you want to remove.

Element 2: Group

***NONE:** The job that you are removing is not a member of a group.

group-name: Specify the name of the group for the job that you are removing.

Element 3: Group sequence

***NONE:** The job that you are removing does not have a sequence number.

group-sequence-number: Specify the sequence number of the job within the group.

Example for RMVJOBJS

Example 1: Removing a Job

```
RMVJOBJS JOB(JOB02 ACCTG 11)
```

In the example, JOB02 which is sequence number 11 in the group ACCTG, is removed.

Error messages for RMVJOBJS

None

RMVJRNCHG (Remove Journaled Changes) Command Description

RMVJRNCHG Command syntax diagram

Purpose

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

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

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

If an error is found at any point while the journal entries are being removed, the operation ends and the objects may be only partially updated from the journal entries.

Additionally, the command can end when journal entries list operations which cannot be replayed by the command. For example, the command ends when a journal entry is found that indicates one of the following has occurred:

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

See the » Journal management article in the Information Center « for a complete listing of the various entries and how they are handled by this command including those entries which can stop the command.

The command also ends on illogical conditions. If the command ends due to illogical conditions and it is logically possible to restart the operation, you can issue the command again, specifying a new starting sequence number.

Note: If the operation ends for one of the objects specified, it ends for all of the objects.

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

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR and QSRV user profiles have private authorities to use the command.
2. The objects specified on this command must currently be having their changes journaled, and they must have been journaled to the specified journal throughout the period indicated on the command.
3. Before images are required for the objects. See the appropriate start journaling command.
4. The objects indicated on the command are allocated exclusively while the changes are being removed. If an object cannot be allocated exclusively, the command ends and no journaled changes are removed from the objects.
5. If there is no journal entry that represents the entry specified on the FROMENT or TOENT parameter the command ends, and no journaled changes are removed from the objects.
6. If the journal sequence numbers have been reset in the range of the receivers specified, and a sequence number is specified on the FROMENT or TOENT parameter, the first occurrence of the sequence number specified on either parameter is used.
7. The TOJOBO parameter cannot be used to specify when the remove journaled changes operation is to end if one or more journal receivers in the specified receiver range was attached to a journal that had » a RCVSIZOPT or FIXLENDTA option specified that omitted the collection of that data. «
8. This command cannot be used on or with a remote journal.

9. The maximum number of objects that can have changes removed with this command is 65,535. If more than 65,535 objects are included in the specifications, an error message is sent and no changes are removed. You can change the values specified on this parameter so that the limit is not exceeded.

Required Parameters

JRN Specifies the qualified name of the journal that contains the journal entries being removed.

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

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

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

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

journal-name: Specify the name of the journal.

FILE Specifies a maximum of 300 qualified names of the physical database files whose journal entries are being removed. The name of the member in the file whose journal entries are being removed can also be specified.

Either the FILE parameter must be specified or the Object Information parameter (OBJ) must be specified, but not both.

Element 1: File Name

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

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

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

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

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

file-name: Specify the name of the physical database file whose journal entries are being removed.

Element 2: Member Name

***FIRST:** Journal entries are removed from the first member of the file.

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

member-name: Specify the name of the member in the file whose journal entries are removed.

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, journal changes are removed from the first member of all applicable files in the library.

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

Either the FILE parameter must be specified or the Object Information parameter (OBJ) must be specified, but not both.

Element 1: Object

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

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

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

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

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

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

Element 2: Object Type

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

***DTAARA:** Entries for data areas are removed.

Element 3: Member Name

<u>*FIRST:</u> Journal entries are removed from the first member of the file.

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

member-name: Specify the name of the member in the file whose journal entries are removed.

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, journal changes are removed from the first member of all applicable files in the library.

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

Optional Parameters

RCVRNG

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

Note: The maximum number of receivers that can be included in a range of receivers is 256. If more than 256 receivers are included in the range specified, an error message is sent and no changes are removed. You can change the values specified on this parameter so that the limit is not exceeded.

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

Element 1: Starting Journal Receiver

The name of the starting journal receiver can be qualified by one of the following library values:

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

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

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

starting-journal-receiver: Specify the name of the journal receiver used as the first (newest) receiver with journal entries to be removed.

Element 2: Ending Journal Receiver

The name of the ending journal receiver can be qualified by one of the following library values:

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

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

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

ending-journal-receiver: Specify the name of the journal receiver used as the last (oldest) receiver with the journal entries being removed. If the end of the receiver chain is reached before a receiver of this name is found, the operation ends.

FROMENT

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

***LAST:** Journal entries are removed starting with the last journal entry in the specified receiver range. If FROMENT is not specified, *LAST is assumed.

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

The system also verifies that the date and time of the saved version of the object that is restored on the system is the same as the date and time that the object was last saved, as indicated on the journal.

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

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

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

starting-sequence-number: Specify the sequence number of the first journal entry that is processed when removing journal changes from the specified object.

TOENT

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

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

ending-sequence-number: Specify the sequence number of the last journal entry that is removed from the specified object.

TOJOBO

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

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

Only objects of type *FILE have journal entries related to job opens.

The job identifier has a maximum of three elements.

```
job-name
user-name/job-name
job-number/user-name/job-name
```

The job name must be specified. The null value (*N) may be used in place of the job number or the user profile name to maintain the position in the sequence. For example, 123456/*N/*job-name* specifies the job number, 123456, and the job name, without specifying the user profile under which the job is run.

CMTBDY

Specifies whether commitment boundaries are honored when the journal entries from which journaled changes are to be removed are part of a commitment control logical unit of work (LUW). More information on the use of commitment control is in the » Commitment control article in the Information Center. «

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

***NO:** The journal entries are removed from the entry specified on the FROMENT parameter to the entry indicated on the TO option, regardless of commitment boundaries. Even if a journal entry within this range is a participant of the LUW, the operation is attempted. The FROMENT parameter must identify a point that is at a commitment boundary.

***YES:** The journal entries are removed from the entry specified on the FROMENT parameter to the entry indicated on the TO option, honoring commitment boundaries.

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

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

Example for RMVJRNCHG

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

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

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

Error messages for RMVJRNCHG

***ESCAPE Messages**

CPF70CC
Cannot perform operation beyond journal entry &7.

CPF70CD
Cannot perform operation beyond journal entry &7.

CPF70CE
Cannot perform operation beyond journal entry &7.

CPF70EB
Referential constraint error on member &3.

CPF70EC
Referential constraint error. Reason code &9.

CPF70EE
Maximum encoded vector access paths for member &3.

CPF7002
File &1 in library &2 not a physical file.

CPF7003
Entry not journaled to journal &1. Reason code &3.

CPF7006
Member &3 not found in file &1 in &2.

CPF7007
Cannot allocate member &3 file &1 in &2.

CPF701B
Journal recovery of an interrupted operation failed.

CPF704A
Record length incorrect for member &3.

CPF704F
TOJOBO or TOJOBC parameter not valid for receiver range.

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

CPF7042
Object not journaled or journaled to different journal.

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

CPF7045
Journal receiver &1 in &2 partially damaged.

CPF7046
Duplicate key not allowed for member &3.

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

CPF7048
Cannot perform journaled change to member &3.

CPF7049
Cannot perform operation beyond journal entry &7.

CPF705A
Operation failed due to remote journal.

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

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

CPF7054
FROM and TO values not valid.

CPF7055
Maximum number of objects exceeded.

CPF7056
Object &1 not being journaled with before images.

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

CPF7058
Apply or remove journaled entries operation failed.

CPF7067
FROMENT option not valid. Commit boundary violation.

CPF7068
Entry needed for apply or remove operation not found.

CPF7069
No entries applied or removed using journal &1.

CPF7077
Key mapping error on member &3.

CPF7078
Cannot apply or remove changes to member &3.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9809
Library &1 cannot be accessed.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

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

RMVLANADPT (Remove LAN Adapter) Command Description

RMVLANADPT Command syntax diagram

Purpose

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

Restrictions:

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

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

Attention:

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

Required Parameters

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

ADPTNAME
Specifies the name of the adapter being removed.

***ADPTADR:** The adapter address is used to identify the adapter.

adapter-name: Specify the name of the adapter being removed. The name can be a maximum of 10 characters in length.

Optional Parameter

ADPTADR
Specifies the 12-character hexadecimal adapter address.

Example for RMVLANADPT

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

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

Error messages for RMVLANADPT

*ESCAPE Messages

CPF8B44
Adapter address required.

CPF8B47
Request to remove adapter &29 on line &23 failed.

CPF8B49
Unable to remove network adapter &29 on line &23.

CPF8B50
Adapter &29 on line &23 cannot be removed.

CPF8B68
Line description &23 not found.

CPF8B70
Network line description &23 not varied on.

CPF8B74
Request to display active adapters failed.

CPF8B75
No adapter entries in network adapter file.

CPF8B76
No functional addresses for adapter.

Printed in U.S.A.