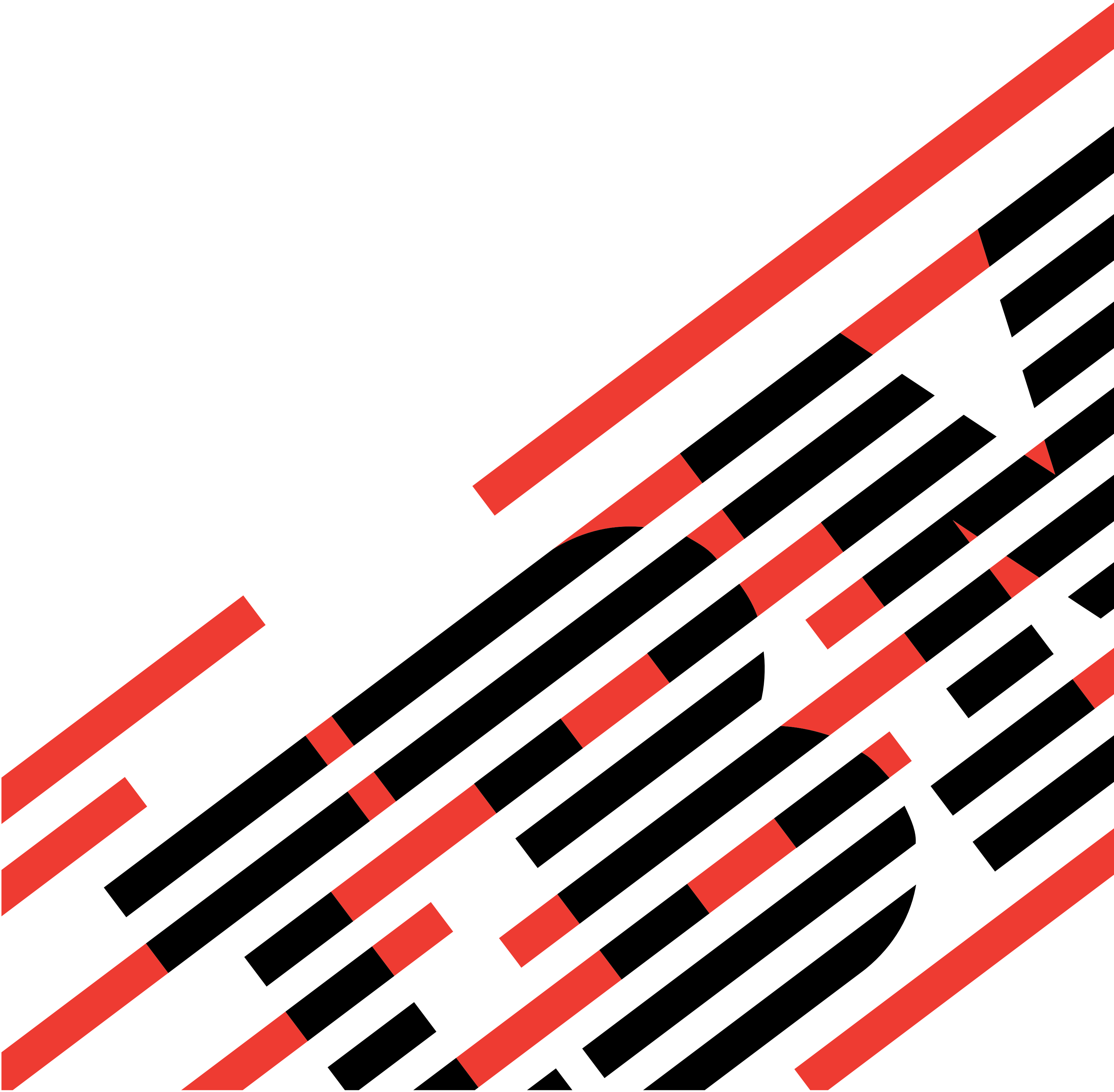
IBM

eserver

iSeries

CL Commands Volume 19



iSeries

CL Commands Volume 19

© Copyright International Business Machines Corporation 1998, 2002. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 1998, 2002

Command Descriptions

RUNSQLSTM (Run Structured Query Language Statement) Command Description

RUNSQLSTM Command syntax diagram

Purpose

The Run Structured Query Language Statement (RUNSQLSTM) command processes a source file of SQL statements.

Parameters

SRCFILE

Specifies the qualified name of the source file that contains the SQL statements to be run.

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

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

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

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

source-file-name: Specify the name of the source file that contains the SQL statements to be run. The source file can be a database file or an inline data file.

SRCMBR

Specifies the name of the source file member that contains the SQL statements to be run.

COMMIT

Specifies whether SQL statements in the source file are run under commitment control.

***CHG or *UR:** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

***ALL or *RS:** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

***CS:** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

***NONE or *NC:** Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP COLLECTION statement is included in the program, *NONE or *NC must be used. If a relational database is specified on the RDB parameter and the relational database is on a system that is not on an iSeries 400, *NONE or *NC cannot be specified.

© Copyright IBM Corp. 1998, 2002

***RR:** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

NAMING

Specifies the naming convention used for naming objects in SQL statements.

***SYS:** The system naming convention (library-name/file-name) is used.

***SQL:** The SQL naming convention (collection-name.table-name) is used.

PROCESS

Specifies whether SQL statements in the source file member are executed or syntax-checked only.

***RUN:** Statement are syntax-checked and run.

***SYN:** Statements are syntax-checked only.

ALWCPYDTA

Specifies whether a copy of the data can be used in a SELECT statement.

***OPTIMIZE:** The system determines whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which method provides the best performance. If COMMIT is *CHG or *CS and ALWBLK is not *ALLREAD, or if COMMIT is *ALL or *RR, then a copy of the data is used only when it is necessary to run a query.

***YES:** A copy of the data is used only when necessary.

***NO:** A copy of the data is not used. If temporary copy of the data is required to perform the query, an error message is returned.

ALWBLK

Specifies whether the database manager can use record blocking, and the extent to which blocking can be used for read-only cursors.

***ALLREAD:** Rows are blocked for read-only cursors if *NONE or *CHG is specified on the COMMIT parameter. All cursors in a program that are not explicitly able to be updated are opened for read-only processing even though EXECUTE or EXECUTE IMMEDIATE statements may be in the program.

Specifying *ALLREAD:

- Allows record blocking under commitment control level *CHG in addition to the blocking allowed for *READ.
- Can improve the performance of almost all read-only cursors in programs, but limits queries in the following ways:
  - The Rollback (ROLLBACK) command, a ROLLBACK statement in host languages, or the ROLLBACK HOLD SQL statement does not reposition a read-only cursor when *ALLREAD is specified.
  - Dynamic running of a positioned UPDATE or DELETE statement (for example, using EXECUTE IMMEDIATE), cannot be used to update a row in a cursor unless the DECLARE statement for the cursor includes the FOR UPDATE clause.

***NONE:** Rows are not blocked for retrieval of data for cursors.

Specifying *NONE:

- Guarantees that the data retrieved is current.
- May reduce the amount of time required to retrieve the first row of data for a query.
- Stops the database manager from retrieving a block of data rows that is not used by the program when only the first few rows of a query are retrieved before the query is closed.
- Can degrade the overall performance of a query that retrieves a large number of rows.

***READ:** Records are blocked for read-only retrieval of data for cursors when:

- *NONE is specified on the COMMIT parameter, which indicates that commitment control is not used.
- The cursor is declared with a FOR FETCH ONLY clause or there are no dynamic statements that could run a positioned UPDATE or DELETE statement for the cursor.

Specifying *READ can improve the overall performance of queries that meet the above conditions and retrieve a large number of records.

ERRLVL

Specifies whether the processing is successful, based on the severity of the messages generated by the processing of the SQL statements. If errors that are greater than the value specified on this parameter occur during processing, no more statements are processed and the statements are rolled back if they are running under commitment control.

10: Statement processing is stopped when error messages with a severity level greater than 10 are received.

severity-level: Specify the severity level to be used.

DATFMT

Specifies the format used when accessing date result columns. For input date strings, the specified value is used to determine whether the date is specified in a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

***JOB:** The format specified for the job is used. Use the Display Job (DSPJOB) command to determine the current date format for the job.

***USA:** The United States date format (mm/dd/yyyy) is used.

***ISO:** The International Organization for Standardization (ISO) date format (yyyy-mm-dd) is used.

***EUR:** The European date format (dd.mm.yyyy) is used.

***JIS:** The Japanese Industrial Standard date format (yyyy-mm-dd) is used.

***MDY:** The date format (mm/dd/yy) is used.

***DMY:** The date format (dd/mm/yy) is used.

***YMD:** The date format (yy/mm/dd) is used.

***JUL:** The Julian date format (yy/ddd) is used.

DATSEP

Specifies the separator used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified on the DATFMT parameter.

***JOB:** The date separator specified for the job is used. Use the Display Job (DSPJOB) command to determine the current value for the job.

'/': A slash (/) is used.

'.': A period (.) is used.

',': A comma (,) is used.

'-': A dash (-) is used.

' ': A blank ( ) is used.

***BLANK:** A blank ( ) is used.

TIMFMT

Specifies the format used when accessing time result columns. For input time strings, the specified value is used to determine whether the time is specified in a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

***HMS:** The hh:mm:ss format is used.

***USA:** The United States time format hh:mm xx is used, where xx is AM or PM.

***ISO:** The International Organization for Standardization (ISO) time format hh.mm.ss is used.

***EUR:** The European time format hh.mm.ss is used.

***JIS:** The Japanese Industrial Standard time format hh:mm:ss is used.

TIMSEP

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when *HMS is specified on the TIMFMT parameter.

***JOB:** The time separator specified for the job is used. Use the Display Job (DSPJOB) command to determine the current value for the job.

':': A colon (:) is used.

'.': A period (.) is used.

',': A comma (,) is used.

' ': A blank ( ) is used.

***BLANK:** A blank ( ) is used.

DECMPT

Specifies the decimal point value used for numeric constants in SQL statements.

***JOB:** The value used as the decimal point for numeric constants in SQL is the representation of decimal point specified by the job running the statement.

***SYSVAL:** The QDECFMT system value is used as the decimal point.

***PERIOD:** A period represents the decimal point.

***COMMA:** A comma represents the decimal point.

SRTSEQ

Specifies the sort sequence table to be used for string comparisons in SQL statements.

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

***LANGIDSHR:** The sort sequence table uses the same weight for multiple characters, and is the shared-weight sort sequence table associated with the language specified on the LANGID parameter.

***LANGIDUNQ:** The unique-weight sort table for the language specified on the LANGID parameter is used.

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

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

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

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

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

table-name: Specify the name of the sort sequence table to be used.

LANGID

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

***JOB:** The LANGID value for the job is retrieved during the precompile.

language-identifier: Specify a language identifier.

DFTRDBCOL

Specifies the collection name used for the unqualified names of tables, views, indexes, and SQL packages. This parameter applies only to static SQL statements.

***NONE:** The naming convention defined on the OPTION parameter is used.

collection-name: Specify the name of the collection identifier. This value is used instead of the naming convention specified on the OPTION parameter.

FLAGSTD

Specifies the American National Standards Institute (ANSI) flagging function. This parameter flags SQL statements to verify whether they conform to the following standards.

```
ANSI X3.135-1992 entry
ISO 9075-1992 entry
FIPS 127.2 entry
```

***NONE:** The SQL statements are not checked to determine whether they conform to ANSI standards.

***ANS:** The SQL statements are checked to determine whether they conform to ANSI standards.

SAAFLAG

Specifies the IBM SQL flagging function. This parameter flags SQL statements to verify whether they conform to IBM SQL syntax More information about which IBM database products IBM SQL syntax is in the *DRDA IBM SQL Reference*, SC26-3255-00.

***NOFLAG:** The SQL statements are not checked to determine whether they conform to IBM SQL syntax.

***FLAG:** The SQL statements are checked to determine whether they conform to IBM SQL syntax.

PRTFILE

Specifies the qualified name of the printer device file to which the RUNSQLSTM printout is directed. The file must have a minimum length of 132 bytes. If a file with a record length of less than 132 bytes is specified, information is lost.

The name of the printer file can be qualified by one of hte 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.

QSYSPRT: If a file name is not specified, the RUNSQLSTM printout is directed to the IBM-supplied printer file QSYSPRT.

printer-file-name: Specify the name of the printer device file to which the RUNSQLSTM printout is directed.

Parameters for SQL procedures:

The parameters listed below only apply to statements within the source file that create SQL procedures. The parameters are used during the creation of the program object associated with the SQL procedure.

TGTRLS

Specifies the release of the operating system on which the user intends to use the object being created.

In the examples given for the *CURRENT value, and when specifying the *release-level* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V2R3M0 is version 2, release 3, modification level 0.

***CURRENT** The object is to be used on the release of the operating system currently running on the user's system. For example, if V2R3M5 is running on the system, *CURRENT means the user intends to use the object on a system with V2R3M5 installed. The user can also use the object on a system with any subsequent release of the operating system installed.

Note: If V2R3M5 is running on the system, and the object is to be used on a system with V2R3M0 installed, specify TGTRLS(V2R3M0) not TGRRLS(*CURRENT).

release-level: Specify the release in the format VxRxMx. The object can be used on a system with the specified release or with any subsequent release of the operating system installed.

Valid values depend on the current version, release, and modification level, and they change with each new release. If you specify a release-level which is earlier than the earliest release level supported by this command, an error message is sent indicating the earliest supported release.

CLOSQLCSR

Specifies when SQL cursors are implicitly closed, SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released. SQL cursors are explicitly closed when you issue the CLOSE, COMMIT, or ROLLBACK (without HOLD) SQL statements.

***ENDACTGRP:** SQL cursors are closed and SQL prepared statements are implicitly discarded.

ENDMOD: SQL cursors are closed and SQL prepared statements are implicitly discarded when the module is exited. LOCK TABLE locks are released when the first SQL program on the call stack ends.

OUTPUT

Specifies whether the precompiler listing is generated.

***NONE:** The precompiler listing is not generated.

***PRINT:** The precompiler listing is generated.

DBGVIEW

Specifies the type of source debug information to be provided by the SQL precompiler.

***NONE:** The source view will not be generated.

***STMT:** Allows the compiled module to be debugged using program statement numbers and symbolic identifiers.

***LIST:** Generates the listing view for debugging the compiled module object.

USRPRF

Specifies the user profile that is used when the compiled program object is run, including the authority that the program object has for each object in static SQL statements. The profile of either the program owner or the program user is used to control which objects can be used by the program object.

***NAMING:** The user profile is determined by the naming convention. If the naming convention is *SQL, USRPRF(*OWNER) is used. If the naming convention is *SYS, USRPRF(*USER) is used.

***USER:** The profile of the user running the program object is used.

***OWNER:** The user profiles of both the program owner and the program user are used when the program is run.

DYNUSRPRF

Specifies the user profile to be used for dynamic SQL statements.

***USER:** For local, dynamic SQL statements run under the user of the program's user. For distributed, dynamic SQL statements run under the profile of the SQL package's user.

***OWNER:** For local, dynamic SQL statements run under the profile of the program's owner. For distributed, dynamic SQL statements run under the profile of the SQL package's owner.

DLYPRP

Specifies whether the dynamic statement validation for a PREPARE statement is delayed until an OPEN, EXECUTE, or DESCRIBE statement is run. Delaying validation improves performance by eliminating redundant validation.

***NO:** Dynamic statement validation is not delayed. When the dynamic statement is prepared, the access plan is validated. When the dynamic statement is used in an OPEN or EXECUTE statement, the access plan is revalidated. Because the authority or the existence of objects referred to by the dynamic statement may change, you must still check the SQLCODE or SQLSTATE after issuing the OPEN or EXECUTE statement to ensure that the dynamic statement is still valid.

***YES:** Dynamic statement validation is delayed until the dynamic statement is used in an OPEN, EXECUTE, or DESCRIBE SQL statement. When the dynamic statement is used, the validation is completed and an access plan is built. If you specify *YES on this parameter, you should check the SQLCODE and SQLSTATE after running an OPEN, EXECUTE, or DESCRIBE statement to ensure that the dynamic statement is valid.

Note: If you specify *YES, performance is not improved if the INTO clause is used on the PREPARE statement or if a DESCRIBE statement uses the dynamic statement before an OPEN is issued for the statement.

Example for RUNSQLSTM

```
RUNSQLSTM  SRCFILE(MYLIB/MYFILE)  SRCMBR(MYMBR)
```

This command processes the SQL statements in member MYMBR found in file MYFILE in library MYLIB.

Error messages for RUNSQLSTM

***ESCAPE Messages**

SQL9006

DB2 Query Manager and SQL Development Kit for AS/400 not at same install level as OS/400.

SQL9010

RUNSQLSTM command failed.

SQL9014

Remote Connection Active

RUNSMGCMD (Run System Manager Command) Command Description

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

RUNSMGCMD Command syntax diagram

Purpose

The Run System Manager Command (RUNSMGCMD) command provides the capability of sending a command from the central site system to one or more managed systems.

Restriction: If a node list (NODL) value is specified, the node list can only contain entries that have a value of *SNA for the address type.

Note: A change request is automatically submitted that can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

Required Parameter

CMD Specifies the command to run. The command can be any command that is run in batch. The command must follow the OS/400 CL command format of 1 to 10 characters with the first character alphabetic and the other nine characters alphanumeric. You can use F4 to enable prompting for a CL command.

Optional Parameters

NODL Specifies the node list object name that contains a list of systems that are the destinations for the request. It cannot be specified if the control point name (CPNAME) parameter is also specified.

<u>*NONE:</u> The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible library values are:

***LIBL:** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

***CURLIB:** The current library for the job is used to locate the node list object.

library-name: Specify that only the library named in this parameter is searched.

node-list-name: Specify the node list object name that contains the list of systems on which the command is to be performed.

CPNAME

Specifies the APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

<u>*NONE:</u> The systems on which this activity is to be performed are not identified individually. A node list must be specified.

***NETATR:** The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

network-identifier: Specify the APPN network identifier of the managed system on which the activity is to be performed.

control-point-name: Specify the APPN control point name of the managed system on which the activity is to be performed.

RMTSTRTIME

Specifies the date and time when the activity can begin running on the managed system. The current date and time values and next date values are determined when the activity begins running at the central site system based on the central site date and time.

Element 1: Time Zone

The time zone of the remote start time.

***LCLSYS:** The remote start time is specified in the time zone of the central site system.

***MGDSYS:** The remote start time is specified in the time zone of the managed system.

Element 2: Start After Time

This is the definition of the time after which the activity is to start.

***CURRENT:** This function can start on the managed system at any time on or after the time this activity is started on the central site system on the date specified in element 3.

start-after-time: Specify the time when this function can start on the managed system. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

Element 3: Start After Date

***CURRENT:** This function starts on the managed system on any date on or after the activity starts on the central site system.

***NEXT:** This function starts on the managed system on any date after the activity starts on the central site system.

start-after-date: Specify the date after the functions start on the managed system. The date must be specified in the job date format.

Element 4: Start Before Time

This element is ignored if start before date is *ANY.

***ANY:** The activity can start at any time on or after the start time.

***CURRENT:** The activity must start before the time when the change request is submitted on the date specified on the start before date element.

start-before-time: Specify the time before which the activity must be started. If the activity cannot start before this time, it never starts. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a time separator (:). With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

Element 5: Start Before Date

***ANY:** The activity can start at any date on or after the start date.

***CURRENT:** The activity must start on the date the change request was submitted.

***NEXT:** The activity must start by the day after the date the change request was submitted.

start-before-date: Specify the date before which the activity must be started. If the activity cannot be started by this date, it never starts. The date must be specified in the job date format.

Note: The special values *CURRENT and *NEXT cannot be specified for the date and time when the time zone value *MGDSYS is specified.

RTNSPLF

Specifies that the output spooled files from the remote command that was submitted returns.

***YES:** All spooled files created from the remote command are returned from the remote system. The spooled files from all the nodes are combined into one spooled file that can be viewed by displaying the command activity details of the change request.

***NO:** The output data is not returned from the managed system.

***FAIL:** The spooled file job log is returned from the managed system if the command fails when running.

USRPRF

Specifies the user profile under which the command runs at the managed system. If the managed system is running the NetView Remote Operations Agent/400 product, this parameter is ignored at the managed system.

***NONE:** No user profile is specified. The default user profile is used on each managed system.

user-profile: Specify the name of the user profile.

PASSWORD

The password for the remote user profile.

- Managed systems at releases prior to V5R1M0 accept only uppercased passwords up to 10 characters long. If a longer password is entered, SMU18A2 message with 100B0007 SNA sense code is returned, indicating that the request was rejected.
- Managed systems at release V5R1M0 and later, running with QPWDLVL system value:
  - 0 or 1 — truncate the received passwords to 10 characters.
  - 2 or 3 — accept passwords up to 128 characters.

***NONE:** No password is specified.

***USRPRF:** The password is the same as the user profile.

password: Specify the password for the user profile.

ENCODE

Specifies whether or not the command, user profile, and password are encoded when sent to the managed systems. If the managed system is running the NetView Remote Operations Agent/400 product, this parameter must be set to *NO.

***NO:** The command, user profile, and password are not encoded when the request is sent to the managed systems.

***YES:** The command, user profile, and password are encoded when the request is sent to the managed systems. The remote command key managed system attribute must be specified on both the central site system and the managed system. This attribute can be changed using the Change Managed System Attributes (CHGMGDSYSA) command. The remote command key must be the same on the central site system and the managed system.

Example for RUNSMGCMD

```
RUNSMGCMD  CMD(STRSBS QSNADS)  NODL(MYLIB/SALESOFC)
  RTNSPLF(*FAIL)
```

Sends a command to start subsystem QSNADS to 20 systems identified in node list SALESOFC. Return the job log information if the command fails.

Error messages for RUNSMGCMD

***ESCAPE Messages**

CPF9838 E
User profile storage limit exceeded.

SMU1604 C
Change request &1 &2 submitted.

SMU1607 E
Not able to process change request &1 &2.

SMU168C D
Node list or control point names required.

SMU16xx E
Start after time of remote stat time not valid.

SMU16xx E
Start before time of remote start time not valid.

SMU16xx E
Start after date of remote start time not valid.

SMU16xx E
Start before date of remote stat time not valid.

SMU16xx E
Remote start time not valid.

«»

RUNSMGOBJ (Run System Manager Object) Command Description

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

RUNSMGOBJ Command syntax diagram

Purpose

The Run System Manager Object (RUNSMGOBJ) command provides the capability of running an object from the central site system to one or more managed systems.

Note: A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

Restrictions:

1. Only OS/400 program objects or file members such as *REXX, *CL, and *UNSPEC can be run.
2. The object cannot reside in QTEMP library.

3. If a node list (NODL) value is specified, the node list can only contain entries that have a value of *SNA for the address type.

Required Parameters

OBJ Specifies the qualified program to be run.

The possible library values are one of the following:

***CURLIB:** The current library for the job is used to locate the object.

library-name: Specify only the library named in this parameter is searched.

object-name: Specify only characters A through Z, and 0 through 9 are used in the object name.

OBJTYPE
Specifies the object to be used.

***PGM:** When this is selected, a program object is to be run.

***FILEDATA:** A CL input stream or *REXX procedure contained in a file is to be run.

***FILE:** The name of the file containing the member with the CL input stream to be run.

Optional Parameters

MBR Specifies the OS/400 physical file member. It is ignored unless the object type is *FILE. (Example: OS/400 CL input stream.)

member-name: Specify the name of the member to be run.

DATATYPE
Specifies the data type of the member. This parameter specifies the type of source file that runs on the managed system.

***UNSPEC:** Unspecified file member type. If the data type cannot be determined at the managed system, or if the name of the file where this member resides is QCLSRC, then the file member is treated as a CL batch input stream. If the source file is named QREXSRC, the file member is treated as a REXX procedure.

***CL:** The file member contains control language, in other words, an OS/400 CL batch input stream.

***REXX:** The file member contains a REXX procedure.

NODL Specifies the node list name that contains a list of systems that are the destinations for the activity. This parameter cannot be specified if the control point name (CPNAME) parameter is also specified.

***NONE:** The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible library values are one of the following:

***LIBL:** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

***CURLIB:** The current library for the job is used to locate the node list object.

library-name: Specify that only the library named in the parameter is searched.

node-list-name: Specify the node list object name that contains the list of systems on which the activity is to be performed.

CPNAME

Specifies the managed systems APPN control point names on which this activity is performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

***NONE:** The systems on which this activity is performed are not identified individually. A node list must be specified.

***NETATR:** The local system network ID is used. This is useful when the node specified is in the same network as the local system.

network-identifier: Specify the managed system APPN network identifier on which the activity is performed.

control-point-name: Specify the managed system APPN control point name on which the activity is performed.

PARM Specifies the parameters to be passed when starting the program. Up to 20 parameters can be specified.

parameter: Specify a 1 to 253 character parameter. The prompt panel initially allows 50 characters to be entered. By entering an ampersand (&) in position 1, the field expands for larger parameters.

Examples for RUNSMGOBJ

Example 1: Running a REXX Procedure

```
RUNSMGOBJ  OBJ(DEVLIB/QREXSRC)   OBJTYPE(*FILEDATA)
  MBR(REPORT1)   CPNAME((*NETATR DEVSYS))
```

This command runs a REXX procedure on the development system.

Example 2: Running a Program

```
RUNSMGOBJ  OBJ(SALESLIB/GENRPTS)   OBJTYPE(*PGM)
  NODL(SYSLIB/ALLSTORES)  PARM(SALES ONLY TODAY)
```

This command runs a program called GENRPTS in library SALELIB on all systems listed in the node list ALLSTORES in library SYSLIB. Two parameters are passed as input to the program. The first parameter has the value SALES ONLY and the second parameter has the value TODAY.

Error messages for RUNSMGOBJ

*ESCAPE Messages

None «

SAV (Save) Command Description

SAV Command syntax diagram

Purpose

The Save (SAV) command saves a copy of one or more objects that can be used in the integrated file system.

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

Restrictions:

1. Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.
2. For detailed restrictions on using this command to save objects in libraries or to save document library objects, see the Backup, Recovery, and Availability topic in the Information Center.

Required Parameter

DEV Specifies the path name of the device to which the objects are saved. See path names for more information on specifying path names.

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

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

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

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

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

Optional Parameters

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

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

Element 1: Object Name

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

'object-path-name': Specify an object path name or a pattern that can match many names.

Element 2: Include or Omit

The second element specifies whether names that match the pattern should be included or omitted from the operation. Note that in determining whether a name matches a pattern, relative name patterns are always treated as relative to the current working directory.

Note: The SUBTREE parameter specifies whether the subtrees are included or omitted.

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

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

SUBTREE

Specifies whether the directory subtrees are included in the save operation.

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

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

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

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

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

SYSTEM

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

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

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

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

CHGPERIOD

Specifies a date and time range. Objects that changed within this range are saved.

Element 1: Starting Date

***ALL:** No starting date is specified. All objects changed prior to the ending date are saved.

***LASTSAVE:** The objects that have changed since the last time they were saved with UPDHST(*YES) specified are saved.

Notes:

1. If this value is specified, the value *ALL must be specified on all other elements of this parameter.
2. For local file systems, the iSeries 400 archive flag is used. For remote file systems, the PC archive flag is used.

start-date: Specify the date after which objects that have changed are saved. The date must be specified in the job date format.

Element 2: Starting Time

***ALL:** All times of day are included in the range.

start-time: Specify the time on the start date after which objects that have changed are saved. The time is specified in 24-hour format with or without a time separator as follows:

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

Note: Specifying an explicit time is valid only if the starting date is an explicit date.

Element 3: Ending Date

***ALL:** No ending date is specified. All objects changed since the starting date are saved.

end-date: Specify the date before which objects that have changed are saved. The date must be specified in the job date format.

Element 4: Ending Time

***ALL:** All times of day are included in the range.

end-time: Specify a time on the end date before which objects that have changed are saved. The time is specified in 24-hour format with or without a time separator as follows:

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

Note: Specifying an explicit time is valid only if the ending date is an explicit date.

TGTRLS

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the *target-release* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level.

Valid values depend on the current version, release, and modification level, and they change with each new release. See the **Valid Values for TGTRLS Parameter** table in the Backup, Recovery, and Availability topic in the Information Center for a complete list of valid values.

***CURRENT:** The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV:** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

target-release: Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

PRECHK

Specifies whether the save operation ends if any of the selected objects cannot be saved.

***NO:** The save operation does not end. Objects that can be saved are saved.

***YES:** The save operation ends. Nothing is saved unless all of the selected objects can be saved.

UPDHST

Specifies whether to update the save history on the objects saved with this save operation. The save history is used when CHGPERIOD(*LASTSAVE) is specified on a later save operation.

***NO:** The save history is not updated.

***YES:** The save history is updated. For local file systems, the system save history is updated. For remote file systems, the PC save history is updated.

***SYS:** The system save history is updated.

***PC:** The PC save history is updated.

VOL Specifies the volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in commonly used parameters.

A maximum of 75 volume identifiers can be specified. After all specified volumes are filled, the save operation continues on whatever volumes are mounted on the device.

***MOUNTED:** The volume currently placed in the device is used. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

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

LABEL

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

***GEN:** The system generates the label name.

- For objects in libraries, this is the equivalent of LABEL(*LIB) on the Save Object (SAVOBJ) and the Save Library (SAVLIB) commands.
- For document library objects, this is the equivalent of LABEL(*GEN) on the Save Document Library Object (SAVDLO) command.
- For objects in other file systems, the label is SAV*yyyymmdd* for a tape and SAV*yyyymmdd*.Q*nnn* for a diskette.

diskette-file-identifier: Specify a maximum of 17 characters as the identifier of the diskette file used for the save operation.

tape-file-identifier: Specify a maximum of 17 characters as the identifier of the tape file used for the save operation.

SEQNBR

Specifies the tape file sequence number to be used.

***END:** The system saves the object starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

sequence-number: Specify the sequence number of the file. Valid values range from 1 through 16777215. Any existing files on the tape at or beyond this sequence number are overwritten.

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

***PERM:** The file is protected permanently.

expiration-date: Specify the date when protection for the file ends.

ENDOPT

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

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

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

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

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

CLEAR

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. Clearing active data removes all files from the volume, starting at the specified sequence number for tape. Replacing active data on optical media replaces only the optical files created by this operation.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. Clearing a diskette does not initialize it. You should initialize diskettes to a save and restore format before the save command is issued by using the Initialize Diskette (INZDKT) command and specifying FMT(*SAVRST).
4. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

<u>*NONE:</u> None of the media is automatically cleared. If the save operation encounters active data on a tape, diskette, or save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL:** All of the media is automatically cleared.

If tapes are used and a sequence number is specified on the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER:** All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape or diskette, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE:** Active data on the media is automatically replaced. Optical volumes are not initialized. Tapes, diskettes, and save files are automatically cleared in the same way as the *ALL value.

DTACPR

Specifies whether data compression is performed.

***DEV:** If the tape device has the hardware compression feature installed, processing proceeds as if DTACPR(*YES) is specified. If the compression feature is not installed or if save data is written to a diskette or save file, processing proceeds as if DTACPR(*NO) is specified.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** No data compression is performed.

***YES:** If the save operation is to tape and the target device has the hardware compression feature, hardware compression is done. If the feature is not present, or if the save data is written to optical, diskette, or save file, software data compression is done. If the save operation is being done while other jobs on the system are active and software data compression is used, the overall system performance may be affected.

COMPACT
Specifies whether device data compaction is performed.

***DEV:** Device data compaction is performed if the data is saved to tape and all tape devices specified on the DEV parameter support the compaction feature.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** Device data compaction is not performed.

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

’*’: The system generates an optical file name in the root directory of the optical volume.

’optical-directory-path-name/’:* The system generates an optical file name in the specified directory of the optical volume.

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

SAVACT
Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state, this parameter is ignored and the save operation is performed as if SAVACT(*NO) was specified.

***NO:** The objects that are in use are not saved. Objects cannot be updated while they are being saved.

***YES:** Objects can be saved and used at the same time. The object checkpoints can occur at different times.

***SYNC:** Objects can be saved and used at the same time. All of the object checkpoints occur at the same time.

SAVACTOPT

Specifies options to be used with the save while active parameter.

***NONE:** No special save while active options will be used.

***ALWCKPWRT:** When *YES or *SYNC is specified for the SAVACT parameter, objects with the QP0L_ATTR_ALWCKPWRT system attribute set will be locked with O_SHRRDWR by the save and can be updated before the save while active checkpoint is reached.

If this option is not specified, normal save while active processing will take place. Objects with the QP0L_ATTR_ALWCKPWRT system attribute set will be locked with O_SHRRD and cannot be updated before the save while active checkpoint is reached.

Note: When *NONE is specified for the SAVACT parameter, *ALWCKPWRT will be ignored.

SAVACTMSGQ

Specifies the path name of the message queue that the save operation uses to notify the user that the checkpoint processing is complete. See path names for more information on specifying path names.

***NONE:** No notification message is sent.

***WRKSTN:** The notification message is sent to the work station message queue.

’message-queue-path-name’: Specify the path name of the message queue.

OUTPUT

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

A stream file or user space is specified as a path name. See path names for more information on specifying path names.

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

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

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

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

INFTYPE

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

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

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

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

USEOPTBLK

Specifies whether or not the optimum block size is used for the save operation.

***YES:** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

If the target release value that is specified is earlier than V3R7M0 then the block size supported by all device types is used.

***NO:** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

» **ASPDEV**

Specifies the name of the auxiliary storage pool (ASP) device to be included in the save operation.

***DFT:** The operation uses the ASPDEV value appropriate for the file system from which objects are being saved. For Integrated File System objects, *ALLAVL is used. For objects from the QSYS file system, the corresponding save command ASPDEV default is used.

***ALLAVL:** The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and all available independent ASPs.

***:** The operation includes the system ASP, all basic user ASPs, and, if the current thread has an ASP group, all independent ASPs in the ASP group.

***SYSBAS:** The system ASP and all basic user ASPs are included in the save operation.

***CURASPGRP:** If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

auxiliary-storage-pool-device-name: Specify the name of the independent ASP device to be included in the save operation. «

Examples for SAV

Example 1: Saving All Data Not in Libraries and Not Document Library Objects

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

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

Example 2: Saving Changes Since the Last Time the Objects Were Saved

```
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
  OBJ(('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT))
  CHGPERIOD(*LASTSAVE)
```

This command saves all objects that are not in libraries, that are not document library objects, and that have changed since the last time they were saved with UPDHST(*YES) specified.

Example 3: Saving Data That Has Not Changed Since 1990

```
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
  OBJ(('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT))
  CHGPERIOD((*ALL *ALL '12/31/90'))
```

This command saves all objects that are not in libraries, that are not document library objects, and that have not changed since December 31, 1990.

Example 4: Saving the Home Directory and Everything Underneath

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

This command saves the user's home directory and everything underneath the home directory to a tape device named TAP01.

Example 5: Saving All Objects in the Current Directory

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

This command uses the default value '*' on the OBJ parameter to save all objects in the current directory and its subdirectories. This example is not valid if the current directory is the root directory or if the current directory is in the QDLS file system.

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

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

Example 6: Omitting Objects During a Save Operation

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

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

```
SAV   DEV('/QSYS.LIB/TAP01.DEVD')  OBJ(('/A')
  ('/A/B/C' *OMIT))
```

This command saves all objects in directory /A and its subdirectories, except those in directory /A/B/C.

Example 7: Saving a Library

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

This command saves library A on the tape device named TAP01.

Example 8: Saving Two Libraries

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

These two commands save two libraries. The first command saves library A and the second command saves library B.

Example 9: Saving All Files in a Library

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

This command saves all files in the library MYLIB on the tape device named TAP01.

Example 10: Saving Two Objects in a Library

```
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
  OBJ('/QSYS.LIB/MYLIB.LIB/MYPGM.PGM')
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
  OBJ('/QSYS.LIB/MYLIB.LIB/MYFILE.FILE')
```

These two commands save two objects in the same library. This first command saves the program MYPGM from library MYLIB. The second command saves the file MYFILE from library MYLIB.

Example 11: Saving a Stream File, a Database File, and a Document

```
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
  OBJ('/MyDir/MyFile')
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
  OBJ('/QSYS.LIB/MYLIB.LIB/MYFILE.FILE')
SAV   DEV('/QSYS.LIB/TAP01.DEVD')
  OBJ('/QDLS/MYFLR/MYDOC')  SUBTREE(*OBJ)
  UPDHST(*YES)
```

Three commands are used in this example to save three objects. The first command saves the stream file MyFile in the directory MyDir on the tape device named TAP01. The second command saves the database file MYFILE in the library named MYLIB on the tape device named TAP01. The third command saves the document MYDOC in a folder named MYFLR on the tape device named TAP01.

Example 12: Saving to a Save File

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

This command saves the directory MYDIR to a save file named MYSAVF.

Example 13: Using Symbolic Links for the Save Operation

```
SAV   DEV('DevLink')
  OBJ(('DirLink') ('FileLink') ('DirLink/*'))
  SAVACT(*YES)  SAVACTMSGQ('MsgqLink')
```

This command assumes that the current directory contains the following symbolic links:

DevLink = /QSYS.LIB/TAP01.DEVD

DirLink = /SomeDirectory

FileLink = /SomeDirectory/SomeFile

MsgqLink = /QSYS.LIB/LIB1.LIB/MSGQ1.MSGQ

This command saves the names associated with DirLink and FileLink, and the objects in SomeDirectory, to device TAP01. A message is sent when the save-while-active checkpoint is complete.

Symbolic links can be used to specify a device, a save-while-active message queue, and an output file. When symbolic links are specified to be saved, only the names of the associated objects are saved, not the content of the associated objects. A symbolic link to a directory can be used to save objects in the directory. Additional information about symbolic links is in the Integrated file system topic in the File systems and management category of the Information Center.

Error messages for SAV

*ESCAPE Messages

CPFA0DB
Object name not a QSYS object.

CPFA0DC
Object name not a QDLS object.

CPFB8ED
Device description &1 not correct for operation.

CPF3708
Save file &1 in &2 too small.

CPF3727
Duplicate device &1 specified on device name list.

CPF3735
Storage limit exceeded for user profile &1.

CPF3738
Device &1 used for save or restore is damaged.

CPF3768
Device &1 not valid for command.

CPF377D
Save ended because of read error on internal system resource.

CPF377E
Not enough storage for save-while-active request.

CPF378A
Message queue not available.

CPF378C
SAVACTMSGQ(*WRKSTN) not valid for batch job.

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

CPF3794
Save or restore operation ended unsuccessfully.

CPF380D
Save or restore of entire system completed unsuccessfully.

CPF3808
Save file &2 in &3 not complete.

CPF381D
Values on CHGPERIOD parameter not valid.

CPF3812
Save file &1 in &2 in use.

CPF382B
Parameters not valid with multiple file systems.

CPF382C
OBJ parameter value not valid for QSYS file system.

CPF382E
Specified parameter not valid for QDLS file system.

CPF382F
OBJ parameter value not valid for QDLS file system.

CPF3823
No objects saved or restored.

CPF3826
*INCLUDE object required on OBJ parameter.

CPF3828
Error occurred while attempting to use &1.

CPF3829
Specified parameter not valid for QSYS file system.

CPF383A
Save or restore ended unsuccessfully.

CPF383B
End of file &1.

CPF383D
Cannot use &1.

CPF3833
Specified value on DEV parameter not valid.

CPF3834
Too many values specified on the DEV parameter.

CPF3835
Tape devices do not support same densities.

CPF3837
&1 objects saved. &2 not saved.

CPF3838
&1 objects saved. &2 objects not saved.

CPF384A
Volume identifier &1 not valid.

CPF384B
Optical file specified not valid.

CPF384C
Error occurred during CCSID conversion.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF386B
Not able to save &1.

CPF386C
Not able to save &1.

CPF3894
Cancel reply received for message &1.

CPF5729
Not able to allocate object &1.

CPF9802
Not authorized to object &2 in &3.

CPF9814
Device &1 not found.

CPF9833
*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group.

CPF9825
Not authorized to device &1.

SAVAPARDTA (Save APAR Data) Command Description

SAVAPARDTA Command syntax diagram

Purpose

The Save APAR Data (SAVAPARDTA) command allows the user to save information required for an Authorized Problem Analysis Report (APAR).

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The following user profiles have private authorities to use the command:
   - QPGMR
   - QSYSOPR
   - QSRV
   - QSRVBAS

Required Parameters

PRBID

Specifies the identifier (ID) of the problem for which APAR data is saved.

***NEW:** An open problem log record is created to track this APAR.

problem-ID: Specify the ID of the problem for which APAR data is saved.

Example for SAVAPARDTA

```
SAVAPARDTA   PRBID(*NEW)
```

This command creates an open problem log for which APAR data is saved. The user selects the data to be saved by indicating the choices on a list display. This data is saved in an APAR library.

Error messages for SAVAPARDTA

***ESCAPE Messages**

CPF2182

Not authorized to library &1.

CPF39FA

Problem &1 &2 &3 not found

CPF39FF

SAVAPARDTA command can not be run

CPF39F2

Cannot allocate library &1

CPF39F4

No more APAR data can be saved for this problem

CPF39F5

Query of problem &1 failed

CPF39F6

Problem record could not be created

CPF39F7

Could not create user space in APAR library

CPF39F8

Problem &1 could not be associated with an APAR library

CPF39F9

Problem &1 &2 &3 in use

SAVCHGOBJ (Save Changed Object) Command Description

SAVCHGOBJ Command syntax diagram

Purpose

The Save Changed Objects (SAVCHGOBJ) command saves a copy of each changed object or group of objects located in the same library. When OBJ(*ALL) is specified, objects can be saved from all user libraries or from up to 300 specified libraries. When saving to a save file, only one library can be specified. For database files, only the changed members are saved. Objects or members changed since the specified date and time are saved.

Objects changed since the specified date and time are saved with the following exceptions:

- If OBJJRN(*NO) is specified, objects currently being journaled are not saved, unless journaling was started after the specified date and time. This ensures that changes made to an object before journaling starts are not lost (because they were not journaled in a journal receiver).
- Freed objects (programs, files, journal receivers, and so forth) are not saved.
- User-defined messages, job and output queue definitions, logical file definitions, and data queue descriptions are saved, but the contents of those objects are not saved. Logical file access paths are saved if ACCPTH(*YES) is specified.

Specified objects that were changed and the libraries where they reside remain locked during the save operation.

Saved objects can be restored with the Restore Object (RSTOBJ) command.

To determine the date and time that an object was changed, run the Display Object Description (DSPOBJD) command with DETAIL(*FULL) specified. For database file members that were changed, run the Display File Description (DSPFD) command.

The types of objects that can be saved by this command are listed for the OBJTYPE parameter in Commonly used parameters. The system saves the changed objects by writing a copy of each one on diskettes, tapes, optical volumes, or a save file. The description of each object is changed with the date, time, and place when it was last saved and is controlled by the UPDHST parameter.

Note: This command ignores all file overrides currently in effect for the job except for the save output file.

Restrictions:

1. To use this command, the user must have the special authority *SAVSYS specified in the user profile by the SPCAUT parameter. Otherwise, the user must have object existence authority for each object specified, and execute authority to both the specified library and the save file library. If the user does not have the necessary authority to a specified object, all changed objects except that object are saved.
2. When saving to tape, diskette, or optical device, the user must have use authority to the device description. When saving to a save file, the user must have object operational and add authorities to the save file.
3. When using a media definition, the user must have *USE authority to the media definition and *EXECUTE authority to the media definition library.
4. All diskettes used to save the changed objects must be initialized in the save/restore format.
5. If tape is used, a standard labeled volume designation must be used.
6. No changed object that is being saved can be changed by a job that is running when the save operation occurs unless save-while-active is used.

7. When the contents of a save file are being saved to the same save file by specifying SAVFDTA(*YES), only the description of the save file is saved.
8. When the contents of a save file are saved with SAVFDTA(*YES), the save file must be restored before objects contained in it can be restored.
9. When using the OUTFILE parameter to save to an existing database file, you must have execute authority to the output file library.
10. Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

Required Parameters

OBJ Specifies the names of one or more objects, or the generic names of each group of objects, to check for changes and then to save those objects that changed. All of the objects must be in the library specified on the LIB parameter.

If the OBJTYPE parameter is not specified, all the object types listed in the description of the OBJTYPE parameter are saved, provided they are in the specified library and have the specified names.

***ALL:** All changed objects in the specified libraries are saved, depending on the values specified for the OBJTYPE parameter.

generic-object-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 functions, refer to generic functions.

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

LIB Specifies the name of the library that contains the changed objects to be saved.

» ***ALLUSR**: All user libraries in the auxiliary storage pools (ASPs) defined by the ASPDEV parameter are searched. 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.

generic-library-name:* Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. Up to 300 generic library names can be specified. A generic library name cannot be specified when saving to a save file. For more information on the use of generic functions, refer to generic functions.

library-name: Specify the name of the library to be saved. Up to 300 library names can be specified. Only one library can be specified when saving to a save file.

DEV Specifies the name of the device on which the changed objects are saved. The device name must already be known on the system by a device description.

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

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

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

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

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

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

Optional Parameters

OBJTYPE

Specifies the types of OS/400 system objects whose changes are saved. The object types saved are also the ones saved and restored by the Save Library (SAVLIB), Restore Object (RSTOBJ), and Restore Library (RSTLIB) commands. Data dictionaries and the associated files are saved only by using the SAVLIB command. More information on this parameter is in Commonly used parameters.

***ALL:** Changes to all object types that are specified by name and that are in the specified library are saved. If *ALL is also specified on the OBJ parameter, changes to all objects in the libraries that are of the types shown in the list referred to are saved.

object-type: Specify the value for each of the types of objects that are saved, such as command (*CMD), file (*FILE), or program (*PGM).

OBJJRN

Specifies whether to save objects that are currently being journaled and that have been journaled since the date and time specified on the REFDATE and REFTIME parameters.

***NO:** Objects being journaled are not saved. If journaling was started after the specified date and time, the changed objects or changed database file members are saved. The date and time of the last journal start operation can be shown by using the Display Object Description (DSPOBJD) command.

***YES:** Objects whose changes are entered in the journal are saved.

REFDATE

Specifies the reference date. Objects in the specified libraries that have been changed since this date, and since the time specified in the REFTIME parameter, are saved.

***SAVLIB:** Objects changed since the last running of the Save Library (SAVLIB) command are saved. If the specified library was never saved, a message is issued and the library is not saved, but the operation continues.

reference-date: Specify the reference date. Objects that have been changed since this date are saved. If you specify a date later than the date of the running of this command, a message is issued and the operation ends. The date must be specified in the job date format.

REFTIME

Specifies the reference time. Objects in the specified libraries that have been changed at or after this time are saved.

***NONE:** No time is specified. Objects changed since the date specified on the REFDATE parameter are saved.

reference-time: Specify a time. Objects that have been changed since this time on the specified date are saved. If REFDATE(*SAVLIB) is specified, no reference time can be specified. 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.

VOL Specifies the volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in Commonly used parameters.

***MOUNTED:** The volume currently placed in the device is used. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are put on the device and used. A blank is used as a separator character when listing multiple identifiers.

SEQNBR

Specifies, when tape is used, which sequence number is used as the starting point for saving changed objects.

***END:** The system saves the changed objects starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

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

LABEL

Specifies the name that identifies the data file on the tape or diskette that is used for the save operation. If the LABEL parameter is used on the save command, the same label must be specified on the restore command.

***LIB:** The file label is created by the system using the name of the library specified on the LIB parameter.

data-file-identifier: Specify the data file identifier (a maximum of 17 characters) of the data file used for the save operation. This option is valid only for the single-library save operations. Do not specify *SAVLIB; the system will return an error message if *SAVLIB is specified.

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

***PERM:** The file is permanently protected.

expiration-date: Specify the date when protection for the file ends.

Note: For operations saving objects to diskette, the expiration date specified must be later than the date of the save operation. Otherwise, the unprotected save/restore files may be lost when the next save/restore file is written during the save operation.

ENDOPT

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

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

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

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

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

STRLIB

Specifies the library with which to begin the *ALLUSR save. This parameter is used to recover from ended or failed *ALLUSR save operations.

Note: The user should not attempt to specify LIB(*ALLUSR) on a failed tape. The proper recovery procedure is to start over with a new tape, as if the user is starting the complete save operation again.

***FIRST:** The save operation begins with the first library in alphabetical order.

library-name: Specify the name of the library with which to begin the save.

SAVF Specifies the qualified name of the save file used to contain the save data. The save file must be empty or CLEAR(*ALL) must be specified.

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

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

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

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

save-file-name: Specify the name of the save file used to contain the saved data.

MEDDFN

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

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

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

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

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

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

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

UPDHST

Specifies whether the save history information of each saved object is changed with the date, time, and location of this save operation. The save history information for an object can be displayed by using the Display Object Description (DSPOBJD) command. This information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) are used on the Apply Journaled Changes (APYJRNCHG) command.

***YES:** The last save date, time, and location information is updated in each object saved.

***NO:** The save history information contained in the description of each object saved is not updated.

Note: UPDHST(*NO) should be used for a save operation that is not intended for recovery. For example, if the save data is transmitted, record by record, to another system and the save file then deleted, the user probably does not want to update the save history information.

OMITLIB

Specifies the names of one of more libraries, or the generic names of each group of libraries, to be excluded from the save operation. If a value other than *NONE is specified then up to 300 values can be specified.

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

library-name: Specify the name of the library to be excluded from the save operation. Up to 300 library names can be specified.

generic-library-name:* Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. Up to 300 generic library names can be specified. For more information on the use of generic functions, refer to generic functions.

OMITOBJ

Specifies the objects to be excluded from the save operation. Up to 300 objects or generic object values can be specified.

If the OMITOBJ parameter is not specified, no objects are excluded from the save operation.

*NONE: No objects are excluded from the save operation.

Element 1: Object Name

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

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

***ALL:** Objects in the specified libraries are excluded, depending on the value specified for the object type.

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

object-name: Specify the name of the object that will be excluded from the save command.

Element 2: Object Type

***ALL:** All object types are excluded from the save operation, depending on the value specified for the object name.

object-type: Specify the type of the object to be excluded from the save operation.

» **ASPDEV**

Specifies the name of the auxiliary storage pool (ASP) device to be included in the save operation.

*: The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

***SYSBAS:** The system ASP and all basic user ASPs are included in the save operation.

***CURASPGRP:** If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

auxiliary-storage-pool-device-name: Specify the name of the independent ASP device to be included in the save operation. «

TGTRLS

Specifies the release level of the operating system on which you intend to restore and use the object.

When specifying the *target-release* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R2M0 is version 5, release 2, modification level 0.

Valid values depend on the current version, release, and modification level, and they change with each new release. See the **Values for TGTRLS Parameter** table in the Backup and Recovery book for a complete list of valid values.

To specify that an object be saved for distribution to a system at a different release level than the system on which the save operation is to occur, the procedure differs for program and non-program objects and by the release level on which program objects are created. If, for example, you are saving an object for distribution to a target system running on an earlier release, you have the following choices:

For program objects

- If the program was created at a release level more current than the targeted earlier release, you must (1) create the program again specifying the targeted earlier release, (2) save the program specifying the targeted earlier release, and then (3) restore the program on the target system.
- If the program was created at the same release level as the target system, you can (1) save the program specifying the targeted earlier release and then (2) restore the program on the target system.

For non-program objects

You can (1) save the object specifying the targeted earlier release and then (2) restore the object on the target system.

Notes:

1. If LIB(*ALLUSR) is specified, only the current release can be the target release.
2. Not all objects can be targeted to another release. To find out which objects are supported, see the table in the Backup and Recovery book.
3. You can use the table in the Backup and Recovery book to find out the values that can be specified, and which release is specified when you choose the *CURRENT and *PRV values, on your operating system.

***CURRENT:** The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV:** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

target-release: Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

CLEAR

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. Clearing active data removes all files from the volume, starting at the specified sequence number for tape. Replacing active data on optical media replaces only the optical files created by this operation.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. Clearing a diskette does not initialize it. You should initialize diskettes to a save and restore format before the save command is issued by using the Initialize Diskette (INZDKT) command and specifying FMT(*SAVRST).
4. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

***NONE:** None of the media is automatically cleared. If the save operation encounters active data on a tape, diskette, or save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL:** All of the media is automatically cleared.

If tapes are used and a sequence number is specified on the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER:** All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape or diskette, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE:** Active data on the media is automatically replaced. Optical volumes are not initialized. Tapes, diskettes, and save files are automatically cleared in the same way as the *ALL value.

PRECHK

Specifies whether the save operation for a library ends if all objects specified by this command do not satisfy the following conditions:

- The objects exist
- The objects were not previously found to be damaged
- The objects are not locked by another job
- The requester of the save operation has authority to save the objects

***NO:** The save operation for a library continues, saving only objects that can be saved.

***YES:** Ends the save operation for a library before any data is written if, after all specified objects are checked, one or more objects cannot be saved. If several libraries are specified, the save operation continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and one or more objects in any library to be saved cannot be saved, the save operation ends and no objects are saved.

SAVACT

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state, this parameter is ignored and the save operation is performed as if SAVACT(*NO) was specified.

***NO:** Objects that are in use are not saved. Objects cannot be updated while they are being saved.

***LIB:** Objects in a library can be saved while they are in use by another job. All of the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

Note: Libraries with thousands of objects may be too large for this option.

***SYNCLIB:** Objects in a library can be saved while they are in use by another job. All of the objects and all of the libraries in the save operation reach a checkpoint together and are saved in a consistent state in relationship to each other.

Note: If you specify this value and LIB(*ALLUSR), and many user libraries exist on the system, it can take a long time to reach a checkpoint for all of the objects and libraries in the save operation.

***SYSDFN:** Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Note: Specifying this value eliminates some size restrictions and may enable a library to be saved that could not be saved with SAVACT(*LIB).

SAVACTWAIT

Specifies the amount of time to wait for a commit boundary or an object that is not available, before continuing the save. If an object remains in use for the specified time, the object is not saved. If a commit boundary is not reached in the specified time, the save operation is ended.

120: The system waits up to 120 seconds for a commit boundary or an object before continuing the save operation.

***NOMAX:** No maximum wait time exists.

wait-time: Specify the time (in seconds) to wait for a commit boundary or an object before continuing the save operation.

SAVACTMSGQ

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when SAVACT(*SYSDFN) or SAVACT(*LIB) is specified. When SAVACT(*SYNCLIB) is specified, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

***NONE:** No notification message is sent.

***WRKSTN:** The notification message is sent to the work station message queue.

The name of the message 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.

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

ACCPTH

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it controls only the saving of the access paths. More information on the restoring of saved access paths is in the Backup, Recovery, and Availability topic in the Information Center.

Attention!

If the based-on physical files and the logical files are in different libraries, the access paths are saved.

However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt.

For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

***NO:** Only objects specified on the command are saved. No logical file access paths are saved.

***YES:** The specified physical files and all eligible logical file access paths over them are saved.

Note: Specifying this value does not save the logical files.

SAVFDTA

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

***YES:** The description and contents of the save file are saved.

***NO:** Only the description of a save file is saved.

DTACPR

Specifies whether data compression is performed.

***DEV:** If the tape device has the hardware compression feature installed, processing proceeds as if DTACPR(*YES) is specified. If the compression feature is not installed or if save data is written to a diskette or save file, processing proceeds as if DTACPR(*NO) is specified.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** No data compression or decompression occurs.

***YES:** If the save operation is to tape and the target device has the hardware compression feature, hardware compression is performed. If the feature is not present, or if the save data is written to optical, diskette, or save file, software compression is performed. If the save operation is being done while other jobs on the system are active and software compression is used, the overall system performance may be affected.

COMPACT

Specifies whether device data compaction is performed.

***DEV:** Device data compaction is performed if the data is saved to tape and all tape devices specified on the DEV parameter support the compaction feature.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** Device data compaction is not performed.

OUTPUT

Specifies whether a list with information about the saved objects is created. The information can be printed with the job's spooled output or directed to a database file. The output or spooled print file does not include records for objects not changed.

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

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

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

Note: If OUTPUT(*OUTFILE) is specified, you must specify the database file name on the OUTFILE parameter.

OUTFILE

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

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

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

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

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

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

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

INFTYPE

Specifies the type of information printed or directed to the database file.

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

***ERR:** The list contains information about the command, an entry for each library, and an entry for each object that was not successfully saved.

***LIB:** The list contains an entry for each library requested to be saved.

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

OPTFILE

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

'*': The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/':* The system generates an optical file name in the specified directory of the optical volume.

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

USEOPTBLK

Specifies whether or not the optimum block size is used for the save operation.

***YES:** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

If the target release value that is specified is earlier than V3R7M0 then the block size supported by all device types is used.

***NO:** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Examples for SAVCHGOBJ

Example 1: Saving Changed Files

```
SAVCHGOBJ   OBJ(ORD*)  LIB(DSTPRODLB)  DEV(TAP01)
  OBJTYPE(*FILE)  REFDATE(122290)
```

This command saves all files with names that start with the characters ORD in the library named DSTPRODLB that were changed since December 22, 1990.

Example 2: Saving Description and Data for Save Files

```
SAVCHGOBJ   OBJ(FILE*)  LIB(MYLIB)  DEV(TAP01)
  OBJTYPE(*FILE)  REFDATE(122290)  SAVFDTA(*YES)
```

This command saves all files with names that start with the characters FILE* in the library named MYLIB that were changed since December 22, 1990. It also saves the description and the data for all save files that match this selection criteria.

Error messages for SAVCHGOBJ

*ESCAPE Messages

CPF3702
&1 objects saved; &2 not saved; &9 not included.

CPF3703
&2 &1 in &3 not saved.

CPF3708
Save file &1 in &2 too small.

CPF3709
Tape devices do not support same densities.

CPF3727
Duplicate device &1 specified on device name list.

CPF3728
Device &1 specified with other devices.

CPF3730
Not authorized to &2 &1 in library &3.

CPF3731
Cannot use &2 &1 in library &3.

CPF3733
&2 &1 in &3 previously damaged.

CPF3735
Storage limit exceeded for user profile &1.

CPF3738
Device &1 used for save or restore is damaged.

CPF3745
No record of SAVLIB operation exists for &1.

CPF3746
System date and time earlier than reference date and time.

CPF3747
Object names cannot be specified with more than one library.

CPF3749
Objects from library &2 not saved.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF377D
Save ended because of read error on internal system resource.

CPF377E
Not enough storage for save-while-active request.

CPF377F
Save-while-active request prevented by pending record changes.

CPF3770
No objects saved or restored for library &1.

CPF3774
Not all objects saved from library &3.

CPF3778
Not all objects saved from all libraries.

CPF378A
Message queue not available.

CPF378C
SAVACTMSGQ(*WRKSTN) not valid for batch job.

CPF378E
Library &1 not saved.

CPF3781
Library &1 not found.

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

CPF3789
Only one library allowed with specified parameters.

CPF379E
Not enough storage available to save library &1.

CPF3790
No available space on diskette.

CPF3793
Machine storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3797
Objects from library &3 not saved. Save limit exceeded.

CPF380B
Save cannot be completed at this time.

CPF3812
Save file &1 in &2 in use.

CPF3815
Save file &1 in &2 too small for save operation.

CPF3818
Starting library &1 not found.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3867
Contents of FILEMBR parameter not correct.

CPF3868
FILEMBR specified but OBJTYPE must be *ALL or *FILE.

CPF3871
No objects saved or restored; &3 objects not included.

CPF3892
&2 &1 in &3 not saved.

CPF3894
Cancel reply received for message &1.

CPF5729
Not able to allocate object &1.

CPF9809
Library &1 cannot be accessed.

CPF9812
File &1 in library &2 not found.

» **CPF9814**
Device &1 not found. «

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

» **CPF9833**
*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group. «

» **CPFB8ED**
Device description &1 not correct for operation. «

*STATUS Messages

CPF3770
No objects saved or restored for library &1.

CPF3774
Not all objects saved from library &3.

CPF3871
No objects saved or restored; &3 objects not included.

SAVCFG (Save Configuration) Command Description

SAVCFG Command syntax diagram

Purpose

The Save Configuration (SAVCFG) command saves all configuration and system resource management (SRM) objects without requiring a system in a restricted state. The information saved includes the following:

- Line descriptions
- Controller descriptions
- Device descriptions
- Mode descriptions
- Class-of-service descriptions
- Network interface descriptions
- Network server descriptions
- NetBIOS descriptions
- Connection lists
- Configuration lists
- Hardware resource data
- Token-ring adaptor data

Information saved can be restored with the RSTCFG command.

Restrictions:

1. You must have *SAVSYS special authority to use this command.
2. SAVCFG data cannot be saved to a diskette device.
3. System resource management (SRM) objects are not saved if a Work with Hardware Products (WRKHDWPRD) job is running at the same time.
4. Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

Required Parameter

DEV Specifies the name of the device on which the configuration information is to be saved. The device name must be known on the system by a device description.

***SAVF:** The save operation uses the save file specified on the SAVF parameter.

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

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

tape-device-name: Specify the name of one or more tape devices used for the save operation. If more than one tape device (up to four) is being used, specify the names of the devices in the order they are used. If more than one tape volume is required, the use of more than one tape device allows one tape volume to be rewound and unloaded while the tape on another device is being processed.

Optional Parameters

VOL Specifies the volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in Commonly used parameters.

***MOUNTED:** The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are put on the device and used. Each volume identifier contains a maximum of 6 alphanumeric characters. A blank is used as a separator character when listing multiple identifiers.

SEQNBR

Specifies the sequence number used as the starting point for the save operation when a tape is used.

***END:** The system saves the file starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

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

EXPDATE

The file is permanently protected.

***PERM:** The save and restore files are permanently protected.

expiration-date: Specify the date when the file protection ends.

ENDOPT

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

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

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

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

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

SAVF Specifies the qualified name of the save file used to contain the save data. The save file must be empty or CLEAR(*ALL) must be specified.

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

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

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

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

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

CLEAR

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. Clearing active data removes all files from the volume, starting at the specified sequence number for tape. Replacing active data on optical media replaces only the optical files created by this operation.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. Clearing a diskette does not initialize it. You should initialize diskettes to a save and restore format before the save command is issued by using the Initialize Diskette (INZDKT) command and specifying FMT(*SAVRST).
4. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

***NONE:** None of the media is automatically cleared. If the save operation encounters active data on a tape, diskette, or save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL:** All of the media is automatically cleared.

If tapes are used and a sequence number is specified on the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER:** All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape or diskette, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE:** Active data on the media is automatically replaced. Optical volumes are not initialized. Tapes, diskettes, and save files are automatically cleared in the same way as the *ALL value.

PRECHK

Specifies whether the save configuration operation ends if any of the objects satisfy the following conditions:

- The objects were previously found to be damaged.
- The objects are locked by another job.
- The user does not have authority to save the objects.

***NO:** The save operation continues, saving only configuration and system resource management (SRM) objects that can be saved.

***YES:** The save operation ends before any data is written if any configuration objects or system resource manager objects cannot be saved.

DTACPR

Specifies whether data compression is performed.

***DEV:** If the tape device has the hardware compression feature installed, processing proceeds as if DTACPR(*YES) is specified. If the compression feature is not specified or if save data is written to optical or a save file, processing proceeds as if DTACPR(*NO) is specified.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** No data compression occurs.

***YES:** If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or the save data is written to optical or a save file, software compression is performed. If the save is running while other jobs on the system are active and software compression is used, overall system performance may be affected.

COMPACT

Specifies whether data compaction is performed.

***DEV:** Device data compaction is performed if the data is saved to tape and all tape devices specified on the DEV parameter support the compaction feature.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** Device data compaction is not performed.

OUTPUT

Specifies whether a list with information about the saved objects is created. The information can be printed with the job's spooled output or directed to a database file.

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

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

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

Note: If OUTPUT(*OUTFILE) is specified, you must specify the database file name on the OUTFILE parameter.

OUTFILE

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

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

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

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

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

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

OUTMBR

Specifies the name of the database file member to which the output of the command is directed when *OUTFILE is specified on the OUTPUT parameter.

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

OPTFILE

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

'*': The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/':* The system generates an optical file name in the specified directory of the optical volume.

USEOPTBLK

Specifies whether or not the optimum block size is used for the save operation.

***YES:** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

If the target release value that is specified is earlier than V3R7M0 then the block size supported by all device types is used.

***NO:** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Examples for SAVCFG

Example 1: Saving Objects

```
SAVCFG  DEV(TAP01)  CLEAR(*ALL)
```

This command saves system resource management objects (hardware resource data and token-ring adaptor data) and all configuration objects (including all line, controller, device, mode, class-of-service, and network descriptions, configuration lists, and connection lists). They are saved on the TAP01 tape drive. CLEAR(*ALL) automatically clears all uncleared tapes when they are encountered.

Example 2: Saving Objects to a Specific Tape

```
SAVCFG  DEV(TAP01)  VOL(ABC)
```

This command saves the SRM and configuration objects on the TAP01 tape drive, starting on the tape volume labeled ABC. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be put on the TAP01 tape drive is issued.

Error messages for SAVCFG

***ESCAPE Messages**

CPF3709
Tape devices do not support same densities.

CPF3727
Duplicate device &1 specified on device name list.

CPF3728
Device &1 specified with other devices.

CPF3731
Cannot use &2 &1 in library &3.

CPF3733
&2 &1 in &3 previously damaged.

CPF3737
Save and restore data area &1 not found.

CPF376D
Not all configuration objects saved to save file &3.

CPF376E
Not all configuration objects saved.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

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

CPF3793
Machine storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3812
Save file &1 in &2 in use.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3894
Cancel reply received for message &1.

CPF5729
Not able to allocate object &1.

CPF9809
Library &1 cannot be accessed.

CPF9812
File &1 in library &2 not found.

CPF9845
Error occurred while opening file &1.

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

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

CPF9850
Override of printer file &1 not allowed.

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

CPF9860
Error occurred during output file processing.

SAVDLOBRM (Save Document Library Object using BRM) Command Description

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

SAVDLOBRM Command syntax diagram

Purpose

The Save Document Library Object using BRM (SAVDLOBRM) command saves a copy of the specified document, folders or distribution objects (mail).

Note:

- When a folder is saved, the folder object is saved along with the documents contained in that folder and the subfolders and documents in the subfolders and all successively nested folders and documents.
- Distribution objects (mail) cannot be saved or restored for individual users. Mail can be saved only for all users.

Restrictions:

1. You must have authority to the SAVDLO command.
2. You must have *ALLOBJ or *SAVSYS special authority to use the following parameter combination on this command:
   - DLO(*ALL) FLR(*ANY)
   - Users that do not have *ALLOBJ or *SAVSYS special authority must:
     – Have *ALL authority for each document or folder to be saved.
     – Be enrolled as Document Interchange Architecture (DIA) users.

3. This command cannot be used while another job is running commands such as RCLDLO, SAVDLO, SAVDLOBRM, RSTDLO and RSTDLOBRM because exclusive use of internal objects may have been obtained by these commands.
4. Determining document or folder ownership does not include checking group profiles if one is associated with the specified user profile.

Notes:

1. Do not precede an entry with an asterisk unless that entry is a “special value” that is shown (on the display itself or in the help information) with an asterisk.
2. The SAVDLOBRM command does not support the *MAIL choice found on the SAVDLO command, DLO parameter.

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

Examples for SAVDLOBRM

Example 1: Performing a Save of all DLO’s

```
SAVDLOBRM DLO(*ALL) FLR(*ANY) DEV(TAP01) MEDPCY(FULL)
```

In this example all folders, documents and mail are saved to device TAP01 using the media policy FULL.

Example 2: Saving Objects Changed After a Specific Date

```
SAVDLOBRM DLO(*SEARCH) DEV(TAP01) MEDPCY(FULL)
REFDATE('1/1/99')
```

In this example all documents changed after 1/1/99 are saved to device TAP01.

Error messages for SAVDLOBRM

None

SAVDLO (Save Document Library Object) Command Description

SAVDLO Command syntax diagram

Purpose

The Save Document Library Object (SAVDLO) command saves copies of the following:

- Documents
- Folders
- Distribution objects (mail)

Notes:

1. When a folder is saved, the folder object is saved along with the documents contained in that folder and the subfolders and documents in the subfolders and all successively nested folders and documents. Specific folders can be saved individually using DLO(*FLRLVL).
2. Distribution objects (mail) cannot be saved or restored for individual users. Mail can be saved only for all users.
3. SAVDLO does not require a dedicated system; however, individual objects in use when the save is issued cannot be saved. To ensure all document library objects are saved, run this command when no document or folder activity is occurring on the system.

Restrictions:

1. You must have *ALLOBJ or *SAVSYS special authority to use the following parameter combinations on this command:
   - DLO(*ALL) FLR(*ANY)
   - DLO(*CHG)
   - DLO(*MAIL)
   - DLO(*SEARCH) OWNER(*ALL)
   - DLO(*SEARCH) OWNER(*user-profile-name*)

   where the *user profile name* specified is not the *user profile name* of the user issuing the SAVDLO command.
2. Users that do not have *ALLOBJ or *SAVSYS special authority must:
   - Have *ALL authority for each document or folder to be saved
   - Be enrolled as Document Interchange Architecture (DIA) users
3. Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.
4. Determining document or folder ownership does not include checking group profiles if one is associated with the specified user profile.
5. When using the OUTFILE parameter to save to an existing database file, you must have execute authority to the output file library.

Required Parameters

DLO Specifies the document library objects to save.

***ALL:** All document library objects further qualified by the FLR parameter are to be saved. Specifying DLO(*ALL) FLR(*ANY) saves all document library objects.

***SEARCH:** All documents and folders that meet the specified search criteria are saved. Search criteria are specified by using the following parameters: FLR, CHKFORMRK, CHKEXP, CRTDATE, DOCCLS, OWNER, REFCHGDATE, and REFCHGTIME.

***CHG:** All documents created or changed, all folders created since the last complete save operation, and all mail is saved.

***SYSOBJNAM:** The documents with system object names specified on the SYSOBJNAM parameter are saved.

***DOCL:** The list of documents referred to in a document list specified on the DOCL parameter are saved.

***MAIL:** The distribution objects and documents referred to by a mail log are saved.

***FLRLVL:** The folders specified on the FLR parameter and the documents in the folders are saved. Subfolder are not saved.

document-name: Specify the user-assigned names of the documents that are to be saved. A maximum of 300 documents can be specified. All documents specified must be in the same folder and that folder must be specified on the FLR parameter.

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

***SAVF:** The save file specified on the SAVF parameter is to be used.

diskette-device-name: Specify the name of the diskette device to be used.

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

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

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

Optional Parameters

FLR Specifies the name of the folder to save. If *NONE or *ANY is not specified then a maximum of 300 folders or generic folder names can be specified.

<u>*ANY:</u> Document library objects can be saved from any folder. Consider the following when using the FLR parameter:

- FLR(*ANY) is not valid when one of the following is specified:
  - DLO(*DOCL)
  - DLO(*FLRLVL)
  - DLO(*document-name*)
- FLR(*ANY) is required when one of the following is specified:
  - DLO(*CHG)
  - DLO(*SYSOBJNAM)
  - DLO(*MAIL)
  - DLO(*SEARCH) SRCHTYPE(*ALL)
- When SAVDLO DLO(*ALL) FLR(*ANY) is specified, the following are saved:
  - All documents
  - All folders
  - All distribution objects (mail)

***NONE:** The documents saved are not in any folder. FLR(*NONE) is valid only when one of the following is specified:

- DLO(*ALL)
- DLO(*SEARCH) SRCHTYPE(*DOC)

folder-name: Specify the user-assigned name of the folder from which the documents are to be saved. The folder name can be a maximum of 63 characters in length.

- Folder objects specified here are saved only when DLO(*ALL) or DLO(*FLRLVL) is specified.
- FLR(*folder-name*) is not valid when one of the following is specified:
  - DLO(*SYSOBJNAM)
  - DLO(*MAIL)
  - DLO(*SEARCH) SRCHTYPE(*ALL)
- Only one folder name can be specified when one of the following is specified:
  - DLO(*DOCL)
  - DLO(*SEARCH) SRCHTYPE(*DOC)
  - DLO(*document-name*)

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

SRCHTYPE

Specifies the type of objects for which to search. This parameter is valid only when DLO(*SEARCH) is specified.

***DOC:** Only documents are to be searched and saved.

***ALL:** Documents and folders are to be searched and saved.

CHKFORMRK

Specifies whether documents that have been marked for offline storage are saved. This parameter is valid only when DLO(*SEARCH) SRCHTYPE(*DOC) is specified.

***NO:** All documents that meet the other search values for this save operation are to be saved regardless of whether or not they were marked for offline storage.

***YES:** Only those documents that meet the other search values and are marked for offline storage are saved. Documents may be marked:

- Keep
- Free
- Delete

CHKEXP

Specifies the date for which all documents with an expiration date before this date are to be saved. This date is assigned by the user when the document was created to specify when the document is no longer needed. This parameter is valid only when DLO(*SEARCH) SRCHTYPE(*DOC) is specified.

***NO:** The expiration date is ignored.

***CURRENT:** All documents with an expiration date before today's date are to be saved.

document-expiration-date: Specify a date when all documents that have an expiration date before this date are to be saved.

CRTDATE

Specifies that documents and folders that have a creation date during the time period specified are to be saved. The time period is specified by a starting time and date and an ending time and date. This parameter is valid only when DLO(*SEARCH) is specified.

Element 1: Starting Time

***AVAIL:** Documents and folders created at any time are eligible for selection.

starting-time: Specify the starting time.

When the starting time is used as a search value, the starting date must not be *BEGIN. The starting time must be the same as the value specified on the REFCHGTIME parameter when the REFCHGTIME parameter is specified.

The time is specified in 24-hour format with or without a time separator as follows:

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

Element 2: Starting Date

***BEGIN:** Documents and folders are saved regardless of the creation date associated with the object.

***CURRENT:** The current date is used.

starting-date: Specify the starting date. The date must be specified in the job date format. The start date must be the same as the value specified on the REFCHGDATE parameter when the REFCHGDATE parameter is specified.

Element 3: Ending Time

Documents and folders must have been created before this time to be saved.

***AVAIL:** Documents and folders created at any time are saved.

ending-time: Specify the ending time. When the ending time is to be used as a search value, the ending date must not be *END. See the description of *starting-time* for details about how time can be specified.

Element 4: Ending Date

Documents and folders must have been created on or before this date to be saved.

***END:** Documents created on any date after the starting date are eligible to be saved.

ending-date: Specify the ending date in job date format.

DOCCLS

Specifies the class of documents being saved.

The class is assigned by the user when the document is created. This parameter is valid only when DLO(*SEARCH) SRCHTYPE(*DOC) is specified.

Note: Although document classes are user-assigned, double-byte character set (DBCS) data cannot be specified on this parameter.

***ANY:** The document class is not used to select documents being saved.

document-class: Specify a character string, ranging from 1 through 16 characters in length, that is used to select documents being saved.

OWNER

Specifies the owner of the documents and folders being saved. This parameter is valid only when DLO(*SEARCH) is specified.

***CURRENT:** Documents and folders owned by the user issuing this command are to be saved.

***ALL:** The OWNER parameter is not used to select documents and folders to be saved. *ALLOBJ or *SAVSYS special authority is required when OWNER(*ALL) is specified.

user-profile-name: Specify the name of the user who owns the documents and folders that are to be saved. All documents and folders owned by this user and that meet the other search values specified are to be saved. *ALLOBJ or *SAVSYS special authority is required if the user profile specified is other than the user profile of the user issuing this command.

REFCHGDATE

Specifies the date after which the folders that are created and the documents that are changed or created are to be saved. The change date is updated when the document content or description is changed. This parameter is valid only when DLO(*SEARCH) is specified.

***ANY:** No specific date is specified. Documents are saved regardless of the date they were created or changed. Folders are saved regardless of the date they were created.

***SAVDLOALL:** Folders that have been created and documents that have been created or changed since the last complete save operation are to be saved.

reference-date: Specify the date after which the created folders or the created or changed documents are saved.

REFCHGTIME
Specifies the time, relative to the date specified on the REFCHGDATE parameter, after which the folders that are created and the documents that are changed or created are to be saved. The change time is updated when the document content or description is changed. This parameter is valid only when DLO(*SEARCH) is specified.

***ANY:** No time is specified. The documents are saved regardless of the date they were created or changed. Folders are saved regardless of the date they were created.

reference-time: Specify the time after which the created folders or the created or changed documents are saved. Information on how to specify time is in the CRTDATE parameter description.

SYSOBJNAM
Specifies the system object name. This parameter is valid only when DLO(*SYSOBJNAM) is specified.

***NONE:** The system object name is ignored.

system-object-name: Specify the system object name of the document to be saved. A full 10 characters must be specified. A maximum of 300 names can be specified.

DOCL Specifies a document list that contains the names of documents to be saved. The document list must be in a folder and the folder must be specified on the FLR parameter. This parameter is valid only when DLO(*DOCL) is specified.

The user must have *USE authority to the document list.

Note: The document list must be the result of a local search, not a remote search.

***NONE:** Documents are not to be saved from a document list.

document-list: Specify the document list containing the names of the documents to be saved.

VOL Specifies the volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in Commonly used parameters.

***MOUNTED:** The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

volume-identifier: Specify the identifiers of up to 75 volumes in the order they are placed in the device and used to save the system data.

SEQNBR

Specifies the starting tape sequence number to use for saving the documents library objects.

***END:** The system saves the document library objects starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

sequence-number: Specify the sequence number of the file that is used as a starting point to save the document library objects. Valid values range from 1 through 16777215.

EXPDATE

Specifies the expiration date of the diskette, tape, or optical file being created. The diskette, tape, or optical files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

Notes:

1. This parameter is valid for tape, diskette, and optical files. For save operations to diskette, the expiration date specified must be later than the date of the save operation. Otherwise, the save and restore files whose expiration date has been exceeded may be lost when the next save and restore file is written during the save operation.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

***PERM:** The file is permanently protected.

expiration-date: Specify the date when protection for the file ends.

ENDOPT

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

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

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

***LEAVE:** The tape is not rewound; another save operation can start at the current position on the tape.

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

LABEL

Specifies the name that identifies the data file on the tape or diskette used for the save. If the LABEL parameter is used, the label must be specified on the restore command.

***GEN:** The system generates the label name.

data-file-identifier: Specify the data file identifier that is used as the label for the data file being used for the save operation. Up to 17 characters can be specified.

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

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

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

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

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

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

OMITFLR

Specifies the names of one of more folders, or the generic names of each group of folders, to be excluded from the save operation. If a value other than *NONE is specified then up to 300 values can be specified.

***NONE:** No folders are excluded from the save operation.

folder-name: Specify the name of the folder to be excluded from the save operation.

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

OUTPUT

Specifies whether a list with information about the saved document library objects is created.

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

***OUTFILE:** The output is directed to a database file specified on the OUTFILE parameter. The file must have the same format as database file QAOJSAVO.

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

OUTFILE

Specifies the qualified name of the database file to which the information about the document library objects is directed. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QAOJSAVO in QSYS as a model with the format name of QJSDLO. This parameter is valid only when OUTPUT(*OUTFILE) is specified.

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

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

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

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

database-file-name: Specify the name of the database file to be used.

OUTMBR

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

Element 1: Member to Receive Output

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

member-name: Specify the name of the file member that is to receive 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 existing records in the specified database file member are replaced by the new records.

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

CLEAR

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. Clearing active data removes all files from the volume, starting at the specified sequence number for tape. Replacing active data on optical media replaces only the optical files created by this operation.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. Clearing a diskette does not initialize it. You should initialize diskettes to a save and restore format before the save command is issued by using the Initialize Diskette (INZDKT) command and specifying FMT(*SAVRST).
4. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

***NONE:** None of the media is automatically cleared. If the save operation encounters active data on a tape, diskette, or save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL:** All of the media is automatically cleared.

If tapes are used and a sequence number is specified on the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER:** All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape or diskette, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE:** Active data on the media is automatically replaced. Optical volumes are not initialized. Tapes, diskettes, and save files are automatically cleared in the same way as the *ALL value.

STG Specifies, only for filed documents, whether the system storage that is occupied by the document being saved is kept, deleted, or freed after the save operation is completed.

Note: STG(*DELETE) and STG(*FREE) are not valid when any of the following are specified:

- DLO(*ALL) FLR(*ANY)
- DLO(*SEARCH) CHKFORMRK(*YES)
- DLO(*CHG)
- DLO(*MAIL)
- SAVACT(*YES)

***KEEP:** The storage occupied by the document remains unchanged after the save operation.

***DELETE:** The document object and all search terms are deleted from the system after the save operation.

***FREE:** The document description and search terms remain on the system but the storage occupied by the document is deleted after the save operation.

DTACPR

Specifies whether data compression is performed.

***DEV:** If the tape device has the hardware compression feature installed, processing proceeds as if DTACPR(*YES) is specified. If the compression feature is not installed or if save data is written to a diskette or save file, processing proceeds as if DTACPR(*NO) is specified.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** No data compression is performed.

***YES:** If the save is to tape and the target device has the hardware compression feature, hardware compression is performed. If the feature is not present or if the save data is written to optical, diskette, or save file, software compression is performed. If the save operation is run while other jobs on the system are active and software compression is used, there may be an effect on the overall system performance.

COMPACT

Specifies whether device data compaction is performed.

***DEV:** Device data compaction is performed if the data is saved to tape and all tape devices specified on the DEV parameter support the compaction feature.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** Device data compaction is not performed.

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.

graphic-character-set-identifier: Specify the graphic character set and code page values that were used to create the command parameters. Each value can be up to 3 bytes in length and must be separated by at least one blank.

TGTRLS

Specifies the release level of the operating system on which you intend to restore and use the object.

When specifying the *target-release* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R2M0 is version 5, release 2, modification level 0.

Valid values depend on the current version, release, and modification level, and they change with each new release. See the **Values for TGTRLS Parameter** table in the Backup and Recovery book for a complete list of valid values.

***CURRENT:** The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV:** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

target-release: Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

SAVACT

Specifies whether an object can be updated while it is being saved.

***NO:** Objects are not allowed to be saved if they are in use by another job.

***YES:** Objects are allowed to be changed during the save request.

Note: Some applications update document library objects directly, as the data is supplied to the application rather than saving the updates in a temporary file and then updating the DLOs. DLOs that are being updated directly (typically, those being updated by PC-based applications) will not be saved. See the Backup, Recovery, and Availability topic in the Information Center for more information on using this parameter.

SAVACTWAIT

Specifies the amount of time to wait for an object that is in use before continuing the save. If a lock is not obtained in the specified time, the object is not saved.

120: The system waits up to 120 seconds for an object lock before continuing the save operation.

***NOMAX:** No maximum wait time exists.

wait-time: Specify the time (in seconds) to wait for an object lock before continuing the save operation. Valid values range from 0 through 99,999.

ASP Specifies the number of the auxiliary storage pool (ASP) of the document library object to be saved.

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

auxiliary-storage-pool-number: Specify the ASP to be saved. Only document library objects that reside in the specified ASP are to be saved. All document library objects in other ASPs are ignored. Valid values must designate an existing ASP that contains document library objects. ASP 1 is the system ASP.

Note: Mail that has not been filed and documents that are not in a folder reside in the system ASP.

OPTFILE

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

'*': The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/':* The system generates an optical file name in the specified directory of the optical volume.

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

USEOPTBLK

Specifies whether or not the optimum block size is used for the save operation.

***YES:** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

If the target release value that is specified is earlier than V3R7M0 then the block size supported by all device types is used.

***NO:** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Examples for SAVDLO

Example 1: Performing a Complete Save Operation

```
SAVDLO  DLO(*ALL) FLR(*ANY) DEV(TAP01)
```

This command saves all folders, documents, and mail to the tape device TAP01.

Example 2: Saving All Changes

```
SAVDLO  DLO(*CHG) DEV(TAP01)
```

This command saves all documents created or changed since the last complete save operation, folders created since the last complete save operation, and all mail.

Example 3: Saving An Entire ASP

```
SAVDLO  DLO(*ALL) DEV(TAP01) FLR(*ANY) ASP(2)
```

This command saves all folders and documents in user ASP 2.

Example 4: Saving Objects Changed After a Specific Date

```
SAVDLO  DLO(*SEARCH) DEV(TAP01) OWNER(*ALL)
  REFCHGDATE('01/01/92')
```

This command saves all documents changed or created after 01/01/92. This command is useful for saving changes between backups of the documents. This command is similar to the Save Changed Objects (SAVCHGOBJ) used for other object types.

Example 5: Saving Documents and Folders Changed After a Specific Date

```
SAVDLO  DLO(*SEARCH) DEV(TAP01) SRCHTYPE(*ALL)
  OWNER(*ALL) REFCHGDATE('01/01/92')
```

This command saves all folders created since 01/01/92 and all documents created or changed since 01/01/92.

Example 6: Saving Documents Created After a Specific Date

```
SAVDLO  DLO(*SEARCH) DEV(TAP01)
  CRTDATE((*AVAIL '01/01/92')) OWNER(*ALL)
```

This command saves all documents created or changed since 01/01/92.

Example 7: Saving Documents and Folders Created After a Specific Date

```
SAVDLO  DLO(*SEARCH) DEV(TAP01) SRCHTYPE(*ALL)
  CRTDATE((*AVAIL '01/01/92')) OWNER(*ALL)
```

This command saves all documents and folders created since 01/01/92.

Example 8: Freeing System Storage During the Save Operation

```
SAVDLO  DLO(DOCX) FLR(FOLDERA) DEV(DKT01) STG(*FREE)
```

This command saves the document named DOCX in folder FOLDERA to the diskette device DKT01. As part of the save operation, the system storage that was occupied by the data portion of the document is freed.

Example 9: Saving Folders

```
SAVDLO  DLO(*ALL) DEV(*SAVF) FLR(BILL TOM MARY)
  SAVF(SAVF1)
```

This command saves the folders BILL, TOM, and MARY, and all the documents in these folders.

Example 10: Saving Just the Folder Level of a Folder

```
SAVDLO  DLO(*FLRLVL) FLR(DEC01)
  DEV(*SAVF) SAVF(MYLIB/SAVF1)
```

This command saves folder DEC01 in folder STATUS and all the documents in this folder to save file SAVF1 in library MYLIB. Subfolders in this folder are not saved.

Example 11: Saving Just Mail

```
SAVDLO  DLO(*MAIL) DEV(*SAVF)
```

This command saves all distribution objects and all documents referred to by a mail log.

Error messages for SAVDLO

***ESCAPE Messages**

CPF3728
Device &1 specified with other devices.

CPF3733
&2 &1 in &3 previously damaged.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF3793
Machine storage limit reached.

CPF3812
Save file &1 in &2 in use.

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

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF8AC3
ASP &5 is not valid.

CPF8A47
Internal system objects in use.

CPF90B2
List of folder names not valid with DLO parameter.

CPF90C1
Document list &1 empty.

CPF90C2
Document list &1 not used.

CPF90D7
Documents not saved.

CPF90E1
Folder &1 cannot be used with TGTRLS(*PRV).

CPF90E5
Not authorized to document list &1.

CPF9006
User not enrolled in system distribution directory.

CPF902E
&1 document library objects saved. &10 not saved.

CPF903B
Too many objects for save file.

CPF903C
No document library objects saved.

CPF9030
Owner profile (&1) not found.

CPF9046
No documents found satisfying search specification in folder &1.

CPF9053
Not authorized to requested function.

CPF9056
Not authorized to requested function.

CPF906B
No document library objects saved.

CPF908A
Requester &1 not enrolled.

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

CPF941B
Save data area &1 in use.

CPF9410
&1 document library objects saved. &10 not saved.

CPF9411
Changes for ASP &1 not saved.

CPF9417
CRTDATE value not valid with REFCHGDATE or REFCHGTIME.

CPF9418
SAVF or OPTFILE parameter not valid.

CPF9419
LABEL(*GEN) required.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9831
Cannot assign device &1.

CPF9845
Error occurred while opening file &1.

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

CPF9850
Override of printer file &1 not allowed.

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

CPF9860
Error occurred during output file processing.

CPF9899
Error occurred during processing of command.

SAVFLRLBRM (Save Folder List using BRM) Command Description

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

SAVFLRLBRM Command syntax diagram

Purpose

The Save Folder List using BRM (SAVFLRLBRM) command saves backup or archive folder lists. Lists are groups of folders that you select to include in a user-defined list name. Various parameters are used to further define the save folder list process to BRMS.

Note: You must have authority to the SAVDLO command to use this command.

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

Example for SAVFLRLBRM

Example 1: Saving Changed Folders in a List

```
SAVFLRLBRM FLRL(LISTA) DEV(TAP01 TAP02)
MEDPCY(INCR) SAVTYPE(*INCR)
```

In the example, changed folders in a folder list (LISTA) will be saved to TAP01 and TAP02 using a media policy called INCR.

Error messages for SAVFLRLBRM

None

SAVLIB (Save Library) Command Description

SAVLIB Command syntax diagram

Purpose

The Save Library (SAVLIB) command allows the user to save a copy of a one or more libraries.

When saving to a save file only one library can be specified.

Documents and folders contained in QDOC library can be saved by the Save Document Library Object (SAVDLO) command.

The SAVLIB command saves the entire library; this includes the library description, the object descriptions, and the contents of the objects in the library. For job queues, message queues, output queues, data queues, and logical files, only the object definitions are saved, not the contents. Logical file access paths may be saved, however, by using the ACCPTH parameter. The contents of a save file can be saved by using the Save Save File Data (SAVSAVFDTA) command or by using the SAVFDTA parameter on the SAVLIB command. The libraries and their objects are not affected in the system unless the command specifies that the storage is to be freed. However, unless UPDHST(*NO) is specified, the description of each library and each object is updated with the date, place, and time it was last saved. If a group of libraries is saved by specifying *NONSYS, *ALLUSR, or *IBM for the LIB parameter, the date, time, and place are updated in the history information for a data area in QSYS (data area QSAVLIBALL, QSAVALLUSR, or QSAVIBM).

The types of objects saved by this command are the same as those listed for the OBJTYPE parameter in with the addition of *DTADCT. Certain OS/400 system objects that are not contained in user libraries (such as user profiles) are not saved by this command. They can be saved by the Save System (SAVSYS) or Save Security Data (SAVSECDTA) commands.

Note: This command ignores all file overrides currently in effect for the job, except for the listing file.

Restrictions:

1. To use this command, the user must have either the special authority *SAVSYS specified in the user profile by the SPCAUT parameter, or the user must have:
   - Read authority for, or be the owner of, each library specified and
   - Object existence authority for each object in the library.

   If the user does not have the correct authorities for all of the libraries and objects specified, only those for which the user does have authority are saved.
2. When saving to tape, diskette, or optical volume, the user must have *USE authority to the device description.
3. When saving to a save file, the user must have add authority and use authority to the save file.
4. When using a media definition, the user must have *USE authority to the media definition and *EXECUTE authority to the media definition library.
5. When using the OUTFILE parameter to save to an existing database file, the user must have execute authority to the output file library.
6. All diskettes used to save the libraries should be initialized in the save/restore format. If tape is used, a standard labeled tape volume should be used.
7. No library being saved, or the objects in it, can be updated by a job that is running at the time the save operation occurs unless save-while-active is used.
8. When the contents of a save file are being saved to the same save file by specifying SAVFDTA(*YES), only the description of the save file is saved.
9. When the contents of a save file are saved by specifying *YES on the SAVFDTA parameter, the save file must be restored before objects contained in it can be restored.
10. Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

Required Parameters

LIB Specifies which libraries are saved.

Notes:

1. If the user specifies *ALLUSR or *IBM on this parameter, this command should be run when the specified libraries are not being used. If objects in a library are in use while the library is being saved, the objects are not saved. To ensure a complete save of all libraries, run this command with the system in a restricted state. For example, if SAVLIB LIB(*ALLUSR) is run when the subsystem QSNADS is active, the QAO* files are not saved in library QUSRSYS. To save the QAO* files, end the QSNADS subsystem before running SAVLIB LIB(*ALLUSR).
2. Doing a SAVLIB LIB(*IBM) and then doing a SAVLIB LIB(*ALLUSR) saves the same libraries as a SAVLIB LIB(*NONSYS), but requires two restore commands.

***NONSYS:** All user-created libraries, the QGPL and QUSRSYS libraries, and licensed program libraries such as QRPG and QIDU are saved. All subsystems must be ended by the End Subsystem (ENDSBS) or End System (ENDSYS) command before this option is specified. When *NONSYS is specified, the libraries are saved in alphabetical order on the media.

» ***ALLUSR**: All user libraries in the auxiliary storage pools (ASPs) defined by the ASPDEV parameter are searched. 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.

***IBM:** Saves all system (IBM) libraries except for the following Q libraries: »

```
QDOC        QRCLxxxxx   QSYS2       QUSRINFSKR
QDOCxxxx    QRCYxxxxx   QSYS2xxxxx  QUSRNOTES
QDSNX       QRECOVERY   QS36F       QUSROND
QGPL        QRPLOBJ     QTEMP       QUSRPOSGS
QGPL38      QRPLxxxxx   QUSER38     QUSRPOSSA
QMPGDATA    QSPL        QUSRADSM    QUSRPYMSVR
```

QMQMDATA
QMQMPROC
QPFRDATA
QRCL
QSPLxxxx
QSRV
QSYS
QSYSxxxxx
QUSRBRM
QUSRDIRCL
QUSRDIRDB
QUSRIJS
QUSRRDARS
QUSRSYS
QUSRVI
QUSRVxRxMx

«

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

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

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

generic-library-name:* Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. A maximum of 300 generic library names can be specified. For more information on the use of generic functions, refer to generic functions.

library-name: Specify the names of the libraries to be saved. A maximum of 300 libraries can be specified. » The names QDOC, QDOCxxxx, QRCYxxxxx, QRECOVERY, QRPLOBJ, QRPLxxxxx, QSPL, QSPLxxxx, QSRV, QSYS, QSYSxxxxx, and QTEMP cannot be specified. «

Notes:

1. When several library names are specified, the overall save time is reduced, compared to that required to save objects from each of the libraries by using individual commands.
2. Only one library can be specified when saving to a save file.

DEV Specifies the name of the device on which the libraries are saved. The device name must already be known on the system by a device description.

***SAVF:** The save operation is done using the save file specified by the Save File (SAVF) parameter.

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

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

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

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

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

Optional Parameters

VOL Specifies the volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in commonly used parameters.

***MOUNTED:** The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

volume-identifier: Specify the identifiers of up to 75 volumes in the order they are placed in the device and used to save the system data.

SEQNBR

Specifies, only when a tape is used, which sequence number is used as the starting point for the save operation.

***END:** The system saves the library starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

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

If *NONSYS, *ALLUSR, or *IBM is specified on the LIB parameter, the save operation for the set of libraries begins at the sequence number specified. The first file saved in this set is the QFILE file. The QFILE file contains the list of libraries saved.

LABEL

Specifies the name that identifies the data file on the tape or diskette that is used for the save operation. If the LABEL parameter is used on the save command, the user must specify this label on the restore command.

***LIB:** The file label is created by the system using name of the library specified on the LIB parameter.

data-file-identifier: Specify the data file identifier (up to 17 characters) of the data file used for the save operation. This option is valid only for saving a single library. Do not specify *SAVLIB; the system will return an error message if *SAVLIB is specified.

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

***PERM:** The files are permanently protected.

expiration-date: Specify the date when protection for the file ends.

Note: For operations to diskette, the end date specified must be later than the date of the save operation. Otherwise, the ended save/restore files may be lost when the next save/restore file is written during the save operation.

ENDOPT

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

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

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

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

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

STRLIB

Specifies the library with which to begin the *NONSYS, *IBM, or *ALLUSR save.

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

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

The basic steps for restarting a save operation are:

1. Check the job log to determine the library where the previous save operation failed. Find the last library saved, which is indicated by a successful save completion message.
2. Load the next tape and ensure the tape is initialized.
3. Issue the following command:

```
SAVLIB LIB(*NONSYS) DEV(TAP01) ENDOPT(*LEAVE)
     STRLIB(library-name) ACCPTH(*YES)
     OMITLIB(library-name)
```

   where the *library-name* for the STRLIB and OMITLIB parameters is the last library successfully saved. This starts the save operation on the library after the last successfully saved library. Specify the value for the ACCPTH parameter that was specified on the previous SAVLIB command.

Note: Restoring the system using this set of media requires two RSTLIB SAVLIB(*NONSYS) commands to restore the libraries.

***FIRST:** The save operation begins with the first library in alphabetic order.

library-name: Specify the name of the library with which to begin the save.

SAVF Specifies the qualified name of the save file used to contain the save data. The save file must be empty or CLEAR(*ALL) must be specified.

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

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

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

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

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

MEDDFN

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

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

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

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

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

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

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

UPDHST

Specifies whether the save history information of each saved object is updated with the date, time, and location of the current save operation. The save history information for an object is displayed using the Display Object Description (DSPOBJD) command. The save history information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) are used on the Apply Journaled Changes (APYJRNCHG) command.

***YES:** The last save date, time, and location information is updated for each object saved.

***NO:** The save history information contained in the description of each object saved is not updated.

Note: UPDHST(*NO) should be used for a save operation that is not intended for recovery. For example, if the save data is transmitted, record by record, to another system and the save file is immediately deleted, the user will probably not want to update the save history information.

OMITLIB

Specifies the names of one of more libraries, or the generic names of each group of libraries, to be excluded from the save operation. If a value other than *NONE is specified then up to 300 values can be specified.

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

generic-library-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 libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. See generic names for additional information.

library-name: Specify the name of the library to be excluded from the save operation.

OMITOBJ

Specifies the objects to be excluded from the save operation. Up to 300 objects or generic object values can be specified.

If the OMITOBJ parameter is not specified, no objects are excluded from the save operation.

***NONE:** No objects are excluded from the save operation.

Element 1: Object Name

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

***ALL:** The specified objects are excluded from all libraries that are part of the save operation.

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

library-name: Specify the name of the the library that contains the object to be excluded from the save operation.

***ALL:** Objects in the specified libraries are excluded, depending on the value specified for the object type.

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

object-name: Specify the name of the object that will be excluded from the save command.

Element 2: Object Type

***ALL:** All object types are excluded from the save operation, depending on the value specified for the object name.

object-type: Specify the type of the object to be excluded from the save operation.

» **ASPDEV**

Specifies the name of the auxiliary storage pool (ASP) device to be included in the save operation.

***:** The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

***SYSBAS:** The system ASP and all basic user ASPs are included in the save operation.

***CURASPGRP:** If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

auxiliary-storage-pool-device-name: Specify the name of the independent ASP device to be included in the save operation. «

TGTRLS

Specifies the release level of the operating system on which you intend to restore and use the object.

When specifying the *target-release* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R2M0 is version 5, release 2, modification level 0.

Valid values depend on the current version, release, and modification level, and they change with each new release. See the **Values for TGTRLS Parameter** table in the Backup and Recovery book for a complete list of valid values.

To specify that an object be saved for distribution to a system at a different release level than the system on which the save operation is to occur, the procedure differs for program and non-program objects and by the release level on which program objects are created. If, for example, you are saving an object for distribution to a target system running on an earlier release, you have the following choices:

For program objects

- If the program was created at a release level more current than the targeted earlier release, you must (1) create the program again specifying the targeted earlier release, (2) save the program specifying the targeted earlier release, and then (3) restore the program on the target system.
- If the program was created at the same release level as the target system, you can (1) save the program specifying the targeted earlier release and then (2) restore the program on the target system.

For non-program objects

You can (1) save the object specifying the targeted earlier release and then (2) restore the object on the target system.

Notes:

1. If LIB(*NONSYS), LIB(*ALLUSR), or LIB(*IBM) is specified, only the current release can be the target release.
2. Not all objects can be targeted to another release. To find out which objects are supported, see the table in the Backup and Recovery book.
3. You can use the table in the Backup and Recovery book to find out the values that can be specified, and which release is specified when you choose the *CURRENT and *PRV values, on your operating system.

***CURRENT:** The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV:** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

target-release: Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

CLEAR

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. Clearing active data removes all files from the volume, starting at the specified sequence number for tape. Replacing active data on optical media replaces only the optical files created by this operation.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. Clearing a diskette does not initialize it. You should initialize diskettes to a save and restore format before the save command is issued by using the Initialize Diskette (INZDKT) command and specifying FMT(*SAVRST).
4. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

***NONE:** None of the media is automatically cleared. If the save operation encounters active data on a tape, diskette, or save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL:** All of the media is automatically cleared.

If tapes are used and a sequence number is specified on the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER:** All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape or diskette, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE:** Active data on the media is automatically replaced. Optical volumes are not initialized. Tapes, diskettes, and save files are automatically cleared in the same way as the *ALL value.

PRECHK

Specifies whether the save operation for a library should end if all objects in the library specified by this command do not satisfy the following conditions of the save operation:

- The objects exist,
- They are not found to be damaged,

- They are not locked by another job, and
- The requester of the save operation has authority to save the objects.

Note: When multiple libraries are specified, PRECHK(*YES) applies to each library independently. For example, the failure to save one library does not affect the saving of other libraries. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not pass the preliminary check, the save operation ends and no libraries are saved.

***NO:** Allows the save operation for a library to continue, saving only objects that can be saved.

***YES:** Ends the save operation for a library before any data is written if, after all objects in the specified library are checked, one or more objects cannot be saved. If multiple libraries are specified, the save operation continues with the next library.

SAVACT

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state, this parameter is ignored and the save operation is performed as if SAVACT(*NO) was specified.

***NO:** Objects that are in use are not saved. Objects cannot be updated while being saved.

***LIB:** Objects in a library can be saved while they are in use by another job. All of the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

Note: Libraries with thousands of objects may be too large for this option.

***SYNCLIB:** Objects in a library can be saved while they are in use by another job. All of the objects and all of the libraries in the save operation reach a checkpoint together and are saved in a consistent state in relationship to each other.

***SYSDFN:** Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Note: Specifying this value eliminates some size restrictions and may enable a library to be saved that could not be saved with SAVACT(*LIB).

SAVACTWAIT

Specifies the amount of time to wait for a commit boundary or an object that is in use before continuing the save. If a lock is not obtained in the specified time, the object is not saved. If a commit boundary is not reached in the specified time, the save operation is ended.

120: The system waits up to 120 seconds for a commit boundary or an object lock before continuing the save operation.

***NOMAX:** No maximum wait time exists.

wait-time: Specify the time (in seconds) to wait for a commit boundary or an object lock before continuing the save operation. Valid values range from 0 through 99,999.

SAVACTMSGQ

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when SAVACT(*SYSDFN) or SAVACT(*LIB) is specified. When SAVACT(*SYNCLIB) is specified, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

***NONE:** No notification message is sent.

***WRKSTN:** The notification message is sent to the work station message queue.

The name of the message 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.

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

ACCPTH

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it controls only the saving of the access paths. More information on the restoring of saved access paths is in the Backup, Recovery, and Availability topic in the Information Center.

Attention!

If the based-on physical files and the logical files are in different libraries, the access paths are saved.

However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt.

For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

***NO:** Only objects in the libraries specified on the command are saved. No logical file access paths are saved.

***YES:** The physical files in the specified libraries and all eligible logical file access paths over them are saved.

Note: Specifying this value does not save the logical files.

SAVFDTA
Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

***YES:** The description and contents of the save file are saved.

***NO:** Only the description of a save file is saved.

STG Specifies whether the system storage occupied by the data portion of the specified members (except for save files), modules, programs, service programs, Structured Query Language (SQL) packages, and journal receivers in the library being saved is freed as part of the save operation. Only the data portion of the objects is freed, not the descriptions of the objects.

***KEEP:** The storage occupied by the data portion of the objects being saved is not freed.

***FREE:** The storage occupied by the data portion of the specified objects being saved is freed as part of the saved operation. The storage for all the objects in a library is freed only after all the objects in that library are saved successfully.

Note: To prevent the possible abnormal end of a program, the program being saved must not be running in the system when *FREE is specified.

DTACPR
Specifies whether data compression is performed.

***DEV:** If the tape device has the hardware compression feature installed, processing proceeds as if DTACPR(*YES) is specified. If the compression feature is not installed or if save data is written to a diskette or save file, processing proceeds as if DTACPR(*NO) is specified.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** No data compression is done.

***YES:** If the save operation is to tape and the target device has the hardware compression feature, hardware compression is done. If the feature is not present, or if the save data is written to optical, diskette, or save file, software data compression is done. If the save operation is being done while other jobs on the system are active and software data compression is used, the overall system performance may be affected.

COMPACT
Specifies whether device data compaction is performed.

***DEV:** Device data compaction is performed if the data is saved to tape and all tape devices specified on the DEV parameter support the compaction feature.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** Device data compaction is not performed.

OUTPUT

Specifies whether a list of information about the saved libraries is created. The information can be printed with the job's spooled output or directed to a database file.

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

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

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

Note: If OUTPUT(*OUTFILE) is specified, you must specify the database file name on the OUTFILE parameter.

OUTFILE

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

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

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

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

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

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

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

INFTYPE

Specifies the type of information which is printed or directed to the database file.

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

***ERR:** The list contains information about the command, an entry for each library, and an entry for each object that was not successfully saved.

***LIB:** The list contains library entry for each library requested to be saved.

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

OPTFILE

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

'*': The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/':* The system generates an optical file name in the specified directory of the optical volume.

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

USEOPTBLK

Specifies whether or not the optimum block size is used for the save operation.

***YES:** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

If the target release value that is specified is earlier than V3R7M0 then the block size supported by all device types is used.

***NO:** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Examples for SAVLIB

Example 1: Saving a Library on a Tape Device

```
SAVLIB   LIB(JOE)  DEV(TAP01)
```

This command saves the library named JOE on the tape that is in the tape device named TAP01. The storage occupied by JOE in the system is not freed, because the STG parameter default (*KEEP) was assumed.

Example 2: Saving on Multiple Volumes

```
SAVLIB   LIB(QGPL)  DEV(DKT01)  VOL(ABC DEF GHI)
```

The general purpose library (QGPL) is saved on the diskettes in the device named DKT01. The diskettes used must have the volume names ABC, DEF, and GHI. If the save operation is not finished when diskette ABC is full, a message is issued to the operator asking for volume DEF to be placed in the device.

Example 3: Freeing Storage when Saving Data

```
SAVLIB   LIB(CUSTDATA)  DEV(DKT01)  VOL(CUSTNM CUSTAD)
  STG(*FREE)
```

The library named CUSTDATA is saved on volumes CUSTNM and CUSTAD, which are put in the diskette device DKT01. The storage occupied by the specified members, modules, programs, service programs, SQL packages, and journal receivers in the CUSTDATA library is freed after it is saved.

Example 4: Saving on Multiple Devices Serially

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

The library named USRLIB is saved on four tape volumes on three tape devices. The volume named USRA is put on the device named TAP01, the volume named USRB on the device named TAP02, the volume named USRC on the device named TAP03, and the volume named USRD on the device named TAP01. The volume named USRA is rewound, and must be unloaded by the operator when processing is complete so that the device named TAP01 can be used for the volume named USRD.

Example 5: Saving on Multiple Devices in Parallel

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

The library named USRLIB is saved on the devices specified in the media definition named MEDDFNA in library LIBA. For information about creating and using a media definition, see the Backup, Recovery, and Availability topic in the Information Center.

Example 6: Saving a Library with a Media File Label

```
SAVLIB   LIB(LIB1)  DEV(TAP01)  LABEL(MONDAYBACKUP)
```

This command uses the tape device named TAP01 to save the library named LIB1 on tape. The library is saved with a media file label of MONDAYBACKUP. This label must be specified when restoring the library or any of its objects.

Example 7: Specifying Where the Save Operation Starts

```
SAVLIB   LIB(*NONSYS)  DEV(TAP01 TAP02)
  STRLIB(MIKESLIB)
```

This command saves all user-created libraries, licensed program libraries, and the QGPL & QUSRSYS libraries, beginning with the library named MIKESLIB on tape devices named TAP01 and TAP02. The libraries are saved in alphabetic order. Therefore, all libraries beginning with and following the name MIKESLIB are saved.

» Example 8: Saving User Libraries in an Independent ASP Group

```
SAVLIB   LIB(*ALLUSR)  DEV(TAP01)  ASPDEV(*CURASPGRP)
```

This command saves all user libraries on the independent ASPs that are in the ASP group for the current thread. It does not save libraries on the system ASP (ASP number 1) or on basic user ASPs (ASP numbers 2-32). ≪

Error messages for SAVLIB

***ESCAPE Messages**

CPF3701
&1 objects were saved; &2 objects were not saved.

CPF3703
&2 &1 in &3 not saved.

CPF3708
Save file &1 in &2 too small.

CPF3709
Tape devices do not support same densities.

CPF3727
Duplicate device &1 specified on device name list.

CPF3728
Device &1 specified with other devices.

CPF3730
Not authorized to &2 &1 in library &3.

CPF3731
Cannot use &2 &1 in library &3.

CPF3733
&2 &1 in &3 previously damaged.

CPF3735
Storage limit exceeded for user profile &1.

CPF3738
Device &1 used for save or restore is damaged.

CPF3749
Objects from library &2 not saved.

CPF3751
Some libraries not saved.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF377D
Save ended because of read error on internal system resource.

CPF377E
Not enough storage for save-while-active request.

CPF377F
Save-while-active request prevented by pending record changes.

CPF3770
No objects saved or restored for library &1.

CPF3771
&1 objects saved from &3. &2 not saved.

CPF3777
Not all libraries saved.

CPF378A
Message queue not available.

CPF378C
SAVACTMSGQ(*WRKSTN) not valid for batch job.

CPF378E
Library &1 not saved.

CPF3781
Library &1 not found.

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

CPF3785
Not all subsystems ended.

CPF3789
Only one library allowed with specified parameters.

CPF379E
Not enough storage available to save library &1.

CPF3790
No available space on diskette.

CPF3793
Machine storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3797
Objects from library &3 not saved. Save limit exceeded.

CPF380B
Save cannot be completed at this time.

CPF3812
Save file &1 in &2 in use.

CPF3815
Save file &1 in &2 too small for save operation.

CPF3818
Starting library &1 not found.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3871
No objects saved or restored; &3 objects not included.

CPF3892
&2 &1 in &3 not saved.

CPF3894
Cancel reply received for message &1.

CPF5729
Not able to allocate object &1.

CPF9809
Library &1 cannot be accessed.

CPF9812
File &1 in library &2 not found.

» **CPF9814**
Device &1 not found. «

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

» **CPF9833**
*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group. «

» **CPFB8ED**
Device description &1 not correct for operation. «

OPT1332
Optical volume not found in device &1.

*STATUS Messages

CPF3770
No objects saved or restored for library &1.

CPF3771
&1 objects saved from &3. &2 not saved.

CPF3871
No objects saved or restored; &3 objects not included.

SAVLIBBRM (Save Library using BRM) Command Description

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

SAVLIBBRM Command syntax diagram

Purpose

The Save Library using BRM (SAVLIBBRM) command saves a copy of one or more libraries.

This command saves the entire library, including the library description, the object descriptions, and the contents of the objects in the library. For job queues, message queues, output queues, and logical files, only the object definitions are saved, not the contents.

Restrictions:

- » You must have authority to the SAVLIB or SAVCHGOBJ commands to use this command.

«

- » You must have *USE authority to any auxiliary storage pool device specified for the ASPDEV parameter.
  «
- When saving to a save file, only one library can be specified.
- No library (or object in the library) being saved can be updated by a job that is running at the same time the save operation occurs unless save-while-active is used.
- This command should not be used by control group *EXIT item processing as results will be unpredictable.

Note: Do not precede an entry with an asterisk unless that entry is a “special value” that is shown (on the display itself or in the help information) with an asterisk.

Examples for SAVLIBBRM

Example 1: Saving a Library on a tape device

```
SAVLIBBRM LIB(MBALIB) DEV(TAP01) MEDPCY(FULL)
```

In this example a library called MBALIB is saved to device TAP01 using the media policy FULL.

Example 2: Saving a Generic Library on Multiple Devices

```
SAVLIBBRM LIB(MBA*) DEV(TAP01 TAP02) MEDPCY(FULL)
```

In this example all libraries are saved that begin with the characters MBA to devices TAP01 and TAP02 using the media policy FULL.

Error messages for SAVLIBBRM

None

SAVLICPGM (Save Licensed Program) Command Description

SAVLICPGM Command syntax diagram

Purpose

The Save Licensed Program (SAVLICPGM) command saves a copy of the objects that make up a licensed program. It saves the licensed program in a form that can be restored by the Restore Licensed Program (RSTLICPGM) command.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. Some licensed programs can only be saved if the user is enrolled in the system distribution directory. See the publication for each licensed program for a description of this restriction.
3. This command does not save code or language objects for the base OS/400 system.

Required Parameters

LICPGM

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

DEV Specifies the name of the device on which the objects are saved. The device name must already be known on the system by a device description.

Up to four tape device names can be specified. Only one save file name can be specified and only one media library device name can be specified.

***SAVF:** The product is saved in the save file specified on the SAVF parameter.

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

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

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

Optional Parameters

OPTION

Specifies which of the optional parts of the licensed program given in the LICPGM parameter are saved.

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

number-of-licensed-program-option: Specify the number of the optional part of the listed licensed program that is saved.

RLS Specifies which version, release, and modification level of the product is saved.

***ONLY:** Only one version, release, and modification level is installed for the product option.

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

LNG Specifies the language saved for the product specified.

Note: This parameter is ignored when OBJTYPE(*PGM) is specified.

***PRIMARY:** The primary language is saved. The primary language is the language of the operating system.

***ALL:** All languages for this product option are saved.

feature-code: Specify the language feature identifier for the language that is saved for the product.

OBJTYPE

Specifies the type of product objects being saved.

***ALL:** Program and language objects specified on the LNG parameter are saved.

***PGM:** Only the program objects for the product are saved.

***LNG:** The objects associated with the language identified on the LNG parameter are saved.

CHKSIG

Specifies if the digital signatures of objects being saved with the licensed program are to be checked.

***SIGNED:** Objects with digital signatures are checked. Objects that are not digitally signed will not have their signatures verified. Any signed object with a signature that is not valid will be identified in a message in the job log and the save will fail.

***ALL:** All objects that can be digitally signed are checked. Any object that can be signed but has no signature will be identified in a message in the job log. The save will not be terminated for unsigned objects. Any signed object with a digital signature that is not valid will be identified in a message in the job log and the save will fail.

***NONE:** Digital signatures of objects will not be checked.«

VOL Specifies the volume identifiers of the tape volumes on which the product data is saved. The volumes must be placed in the tape device in the same order as the volume identifiers are specified in this parameter. Up to 75 entries can be specified.

***MOUNTED:** The licensed program data is saved on whatever volumes are placed in the tape device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

volume-identifier: Specify the identifiers of one or more volumes in the order they are placed in the tape device and used to save the licensed program.

SEQNBR

Specifies which sequence number is used for the save operation.

***END:** The system saves the licensed program starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

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

EXPDATE

Specifies the end date of the tape save/restore files. If a date is specified, the files are protected and cannot be overwritten until the specified expiration date.

***PERM:** The save and restore files are permanently protected.

expiration-date: Specify the date when the save/restore files are no longer protected.

ENDOPT

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

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

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

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

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

SAVF Specifies the qualified name of the save file used to contain the save data. The save file must be empty or CLEAR(*ALL) must be specified.

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

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

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

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

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

TGTRLS

Specifies the release level of the operating system on which you intend to restore and use the product. When specifying the *target-release* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V3R6M0 is version 3, release 6, modification level 0.

Valid values depend on the current version, release, and modification level, and they change with each new release. See the **Valid Values for TGTRLS Parameter** table in the Backup, Recovery, and Availability topic in the Information Center for a complete list of valid values.

Note: This value is limited by the minimum operating system release specified when the product was created.

***CURRENT:** The product is to be restored to, and used on, the release of the operating system currently running on your system. The product can also be restored to a system with any subsequent release of the operating system installed.

***PRV:** The product is to be restored to the previous release with modification level 0 of the operating system. The product can also be restored to a system with any subsequent release of the operating system installed.

target-release: Specify the release in the format VxRxMx. The product can be restored to a system with the specified release or with any subsequent release of the operating system restore.

CLEAR

Specifies whether tapes, diskettes, or save files that contain active data are automatically cleared, or optical volumes are automatically initialized. Active data is any media file that has not expired.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.
2. Clearing an optical volume does initialize it.

***NONE:** None of the media is automatically cleared. If the save operation encounters active data on a tape, diskette, or save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL:** All of the media is automatically cleared.

If tapes are used and a sequence number is specified on the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER:** All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape or diskette, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE:** Active data on the media is automatically replaced. Tapes are automatically cleared in the same way as the *ALL value.

DTACPR

Specifies whether data compression is performed.

***DEV:** If the tape device has the hardware compression feature installed, processing proceeds as if DTACPR(*YES) is specified. If the compression feature is not specified or if save data is written to optical or a save file, processing proceeds as if DTACPR(*NO) is specified.

***NO:** No data compression is performed.

***YES:** Program data compression is performed. If the program is saved to tape and the target device has the hardware compression feature, hardware compression is performed. If the feature is not present or if the product is being saved to an optical or save file, software compression is performed. If the save operation is running while other jobs on the system are active, and software compression is used, there may be an effect on the overall system performance.

» **LICACPRQD**

Specifies whether or not the user will be prompted for license acceptance when the saved licensed program is restored. If the licensed program being saved does not have online software agreements, no prompt will be done at restore time. This parameter does not affect the license acceptance status of the licensed program currently installed.

Note:

1. If the software agreement of the product has not previously been accepted, as will be the case for a product that is being created and is only packaged, then this parameter is ignored.

***NO:** License acceptance will not be required when the saved licensed program is restored.

***YES:** License acceptance will be required when the saved licensed program is restored, if the licensed program being saved has online software agreements.«

Examples for SAVLICPGM

Example 1: Tapes Cleared Automatically

```
SAVLICPGM   LICPGM(5716WP1)  DEV(TAP01)  CLEAR(*ALL)
```

This command saves the *BASE option of the OfficeVision licensed program. It is saved on the tape that is in the TAP01 tape drive. Each uncleared tape is cleared automatically when it is encountered, and the save operation continues without operator intervention.

Example 2: Saving on Labeled Tape Volume

```
SAVLICPGM   LICPGM(5716WP1)  DEV(TAP01)  VOL(ABCDE)
```

The *BASE option of the OfficeVision licensed program is saved on the TAP01 tape drive, starting on the tape volume labeled ABCDE. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be placed in the TAP01 tape drive is shown to the operator.

Example 3: Saving on Multiple Volumes

```
SAVLICPGM   LICPGM(5716SS1)  OPTION(2)  DEV(TAP01 TAP02)
```

The online help option of the OS/400 system is saved on tape drives TAP01 and TAP02 in alternating order. If the save operation exceeds the storage capacity of two tapes, a message requesting that another volume be put on TAP01 is shown to the operator. The tapes are rewound at the completion of the save operation.

Example 4: Saving to Save Files Cleared Automatically

```
SAVLICPGM   LICPGM(5716WP1)  DEV(*SAVF)
  CLEAR(*ALL)  SAVF(MYLIB/MYSAVF)
```

This command saves the base option of the OfficeVision licensed program. It is saved to the save file MYSAVF in the library MYLIB. The save file MYSAVF is cleared automatically and the save operation continues without operator intervention.

Example 5: Saving program objects to a save file

```
SAVLICPGM   LICPGM(1MYPROD)  OPTION(*BASE)
  DEV(*SAVF)  RLS(*ONLY)  OBJTYPE(*PGM)
  SAVF(MYLIB/MYSAVF)
```

The 1MYPROD product consists of two libraries and does not have any folders. Only the program objects for 1MYPROD product will be saved in the save file, MYSAVF, in MYLIB library.

Example 6: Saving all objects in one library to a save file

```
SAVLICPGM   LICPGM(1MYPROD)  OPTION(*BASE)
  DEV(*SAVF)  RLS(*ONLY)  LNG(2924)
  OBJTYPE(*ALL)  SAVF(MYLIB/MYSAVF)
```

The 1MYPROD product consists of only one library and does not have any folders. The program objects and language objects for the NLV 2924 of 1MYPROD product will be saved in the save file, MYSAVF, in MYLIB library.

Example 7: Saving all objects in two libraries to a save file

```
SAVLICPGM   LICPGM(1MYPROD)  OPTION(*BASE)
  DEV(*SAVF)  RLS(*ONLY)   LNG(2924)
  OBJTYPE(*ALL)  SAVF(MYLIB/MYSAVF)
```

The 1MYPROD product consists of two libraries and has one root folder with the *PGM objects, and one subfolder of that root folder with the *LNG objects. The program objects and language objects for the NLV 2924 of 1MYPROD product will be saved in the save file, MYSAVF, in MYLIB library.

Example 8: Saving for a previous release.

```
SAVLICPGM   LICPGM(1MYPROD)  DEV(TAP01)  TGTRLS(*PRV)
```

The 1MYPROD product will be saved to TAP01 and may be restored on a system with the previous release of the operating system installed.

Error messages for SAVLICPGM

***ESCAPE Messages**

CPF37A2
Licensed program &1 not valid.

CPF3728
Device &1 specified with other devices.

CPF3733
&2 &1 in &3 previously damaged.

CPF3884
Licensed program &1 option &2 not processed.

SAVMEDIBRM (Save Media Information using BRM) Command Description

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

SAVMEDIBRM Command syntax diagram

Purpose

The Save Media Information using BRM (SAVMEDIBRM) command saves a copy of the data required to perform recovery of items saved that use BRMS. Recovery data can be saved at the object or library level.

Note: You must have authority to the SAVOBJ command to use this command.

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

Example for SAVMEDIBRM

Example 1: Saving media information for a Control Group

```
SAVMEDIBRM DEV(*MEDCLS) MEDPCY(FULL) OPTION(*OBJ)
CTLGRP(CG01)
```

In this example, object level information is saved for a control group CG01. The recovery information can be written to any device that supports the media class specified in the media policy FULL.

Error messages for SAVMEDIBRM

None

SAVOBJ (Save Object) Command Description

SAVOBJ Command syntax diagram

Purpose

The Save Object (SAVOBJ) command saves a copy of a single object or a group of objects located in the same library. When OBJ(*ALL) is specified, objects can be saved from up to 300 libraries. When saving to a save file, only one library can be specified. The types of objects that can be saved by this command are listed in the OBJTYPE parameter. The system saves the specified objects by writing a copy of each one on diskettes, tapes, optical volumes or in a save file. Objects in the system are not affected unless the command specifies that the storage should be freed. However, the description of each object is changed with the date, time, and place when it was last saved unless UPDHST(*NO) is specified.

For job queues, output queues, data queues, message queues, and logical files, only the object descriptions are saved, and the contents of the objects are not saved. However, logical file access paths can be saved by the use of the ACCPTH parameter. The contents of a save file can be saved by use of the Save Save File Data (SAVSAVFDTA) command or by specifying the SAVFDTA parameter on this command.

Note: This command ignores all file overrides currently in effect for the job, except for the output file.

Restrictions:

1. To use this command, the user must have either the special authority *SAVSYS specified in the user profile by the SPCAUT parameter, or have (a) object existence authority for each object specified, and (b) execute authority for both the specified library and the save file library. If the user does not have the necessary authority to a specified object, all objects except that one are saved.
2. This command does not save the data dictionary for the library or its associated database files. To save them, the SAVLIB command should be used.
3. When saving to tape, diskette, or optical device, the user must have use authority for the device description and device file.
4. When saving to a save file, the user must have add authority and use authority for the save file.
5. When using a media definition, the user must have *USE authority to the media definition and *EXECUTE authority to the media definition library.
6. When using the OUTFILE parameter to save to an existing database file, you must have execute authority to the output file library.
7. All diskettes used to save the objects must be initialized in the save/restore format.
8. If tape is used, a standard-labeled volume must be placed in the device.
9. No object being saved can be changed by a job that is running at the time the save operation occurs unless save-while-active is used.
10. When the contents of a save file are being saved to the same save file by specifying SAVFDTA(*YES), only the description of the save file is saved.
11. When the contents of a save file are saved by specifying *YES on the SAVFDTA parameter, the save file must be restored before objects contained in it can be restored.
12. Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

Required Parameters

OBJ Specifies the names of one or more objects, or the generic name of each group of objects, to save. The objects being saved must be in the library specified in the LIB parameter.

If the OBJTYPE parameter is not specified, all the object types listed in the description of the OBJTYPE parameter are saved, provided they are in the specified library and have the specified names.

***ALL:** Objects in the specified libraries are saved, depending on the values specified for the OBJTYPE parameter.

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

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

LIB Specifies which libraries contain the objects to be saved.

generic-library-name:* Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*)

substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. Up to 300 generic library names can be specified. A generic library name cannot be specified when saving to a save file. For more information on the use of generic functions, refer to generic object names.

library-name: Specify the name of the library to be saved. Up to 300 library names can be specified. Only one library can be specified when saving to a save file.

DEV Specifies the name of the device on which the objects are saved. The device name must already be known on the system by a device description.

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

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

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

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

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

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

Optional Parameters

OBJTYPE

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

***ALL:** All object types specified by name and in the specified library are saved. If *ALL is also specified on the OBJ parameter, all the objects in the library that are of types that can be saved are saved.

object-type: Specify the value for each of the types of objects to be saved.

The object types saved are also the ones saved and restored by the Save Changed Object (SAVCHGOBJ), Save Library (SAVLIB), Restore Object (RSTOBJ), and the Restore Library (RSTLIB) commands. Data dictionaries and the associated files are saved only by using the SAVLIB command.

VOL Specifies the volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in commonly used parameters.

***MOUNTED:** The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are placed in the device and are used to save the objects. A maximum of 75 volume identifiers can be specified.

If more volumes are needed than are specified, the data is saved on the additional volumes placed in the device.

SEQNBR

Specifies, only when tape is used, which sequence number to use for saving the objects.

***END:** The system saves the objects starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

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

LABEL

Specifies the name that identifies the data file on the tape or diskette that is used for the save. If the LABEL parameter is used on the save command, the user must specify this label on the restore command.

***LIB:** The file label is created by the system using the name of the library specified on the LIB parameter.

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

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

Note: This parameter is valid for tape, diskette, and optical devices.

***PERM:** The file is protected permanently.

expiration-date: Specify the date when protection for the file ends.

Note: For save operations to diskette, the expiration date specified must be later than the date of the save operation. Otherwise, the save and restore files whose expiration date has been exceeded may be lost when the next save and restore file is written during the save operation.

ENDOPT

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

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

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

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

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

SAVF Specifies the qualified name of the save file used to contain the save data. The save file must be empty or CLEAR(*ALL) must be specified.

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

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

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

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

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

MEDDFN

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

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

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

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

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

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

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

UPDHST

Specifies whether the save history information of each saved object is changed with the date, time, and location of the save operation. The save history information for an object is displayed using the Display Object Description (DSPOBJD) command. The save history information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) are used on the Apply Journaled Changes (APYJRNCHG) command.

***YES:** The last save date, time, and location information is changed for each object saved.

***NO:** The save history information contained in the description of each object saved are not changed.

Note: UPDHST(*NO) should be used for a save operation that is not intended for recovery. For example, if the save data is sent, record by record, to another system and the save file immediately deleted, the save history information is probably not to be updated.

OMITLIB

Specifies the names of one of more libraries, or the generic names of each group of libraries, to be excluded from the save operation. If a value other than *NONE is specified then up to 300 values can be specified.

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

generic-library-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 libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. For more information on the use of generic functions, refer to generic object names.

library-name: Specify the name of the library to be excluded from the save operation.

TGTRLS

Specifies the release level of the operating system on which you intend to restore and use the object.

When specifying the *target-release* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R2M0 is version 5, release 2, modification level 0.

Valid values depend on the current version, release, and modification level, and they change with each new release. See the **Values for TGTRLS Parameter** table in the Backup and Recovery book for a complete list of valid values.

To specify that an object be saved for distribution to a system at a different release level than the system on which the save operation is to occur, the procedure differs for program and non-program objects and by the release level on which program objects are created. If, for example, you are saving an object for distribution to a target system running on an earlier release, you have the following choices:

For program objects

- If the program was created at a release level more current than the targeted earlier release, you must (1) create the program again specifying the targeted earlier release, (2) save the program specifying the targeted earlier release, and then (3) restore the program on the target system.
- If the program was created at the same release level as the target system, you can (1) save the program specifying the targeted earlier release and then (2) restore the program on the target system.

For non-program objects

You can (1) save the object specifying the targeted earlier release and then (2) restore the object on the target system.

Notes:

1. Not all objects can be targeted to another release. To find out which objects are supported, see the table in the Backup and Recovery book.
2. You can use the table in the Backup and Recovery book to find out the values that can be specified, and which release is specified when you choose the *CURRENT and *PRV values, on your operating system.

***CURRENT:** The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV:** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

target-release: Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

CLEAR

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. Clearing active data removes all files from the volume, starting at the specified sequence number for tape. Replacing active data on optical media replaces only the optical files created by this operation.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. Clearing a diskette does not initialize it. You should initialize diskettes to a save and restore format before the save command is issued by using the Initialize Diskette (INZDKT) command and specifying FMT(*SAVRST).
4. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

***NONE:** None of the media is automatically cleared. If the save operation encounters active data on a tape, diskette, or save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL:** All of the media is automatically cleared.

If tapes are used and a sequence number is specified on the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER:** All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape or diskette, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE:** Active data on the media is automatically replaced. Optical volumes are not initialized. Tapes, diskettes, and save files are automatically cleared in the same way as the *ALL value.

PRECHK

Specifies whether the save operation for a library ends if all objects specified by this command do not satisfy the following conditions:

- The objects exist
- They are not found to be damaged
- They are not locked by another job
- The requester of the save operation has authority to save the objects

***NO:** The save operation for a library continues, saving only objects that can be saved.

***YES:** The save operation for a library ends before any data is written if, after specified objects are checked, one or more objects cannot be saved. If multiple libraries are specified, the save operation continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

SAVACT

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state, this parameter is ignored and the save operation is performed as if SAVACT(*NO) was specified.

***NO:** Objects that are in use are not saved. Objects cannot be updated while being saved.

***LIB:** Objects in a library can be saved while they are in use by another job. All of the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

Note: Libraries with thousands of objects may be too large for this option.

***SYNCLIB:** Objects in a library can be saved while they are in use by another job. All of the objects and all of the libraries in the save operation reach a checkpoint together and are saved in a consistent state in relationship to each other.

***SYSDFN:** Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Note: Specifying this value eliminates some size restrictions and may enable a library to be saved that could not be saved with SAVACT(*LIB).

SAVACTWAIT

Specifies the amount of time to wait for a commit boundary or an object that is in use before continuing the save. If a lock is not obtained in the specified time, the object is not saved. If a commit boundary is not reached in the specified time, the save operation is ended.

120: The system waits up to 120 seconds for a commit boundary or an object lock before continuing the save operation.

***NOMAX:** No maximum wait time exists.

wait-time: Specify the time (in seconds) to wait for a commit boundary or an object lock before continuing the save operation. Valid values range from 0 through 99,999.

SAVACTMSGQ

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when SAVACT(*SYSDFN) or SAVACT(*LIB) is specified. When SAVACT(*SYNCLIB) is specified, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

***NONE:** No notification message is sent.

***WRKSTN:** The notification message is sent to the work station message queue.

The name of the message 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.

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

FILEMBR

Specifies the database file members to be saved. This parameter has two parts: the file name and the member name.

Element 1: Database File Names

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

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

Restrictions:

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

Element 2: Member Names

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

***NONE:** No members are saved from the specified file.

generic-member-name:* Specify the generic name of the member. 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 the name. For more information on the use of generic functions, refer to generic object names.

member-name: Specify the names of the members saved from the specified file.

Restrictions:

- If nongeneric member names are specified, the specified members must exist in the file for any part of the file to be saved or restored.
- If generic member names are specified, the file must contain member names that match the generic names for the file to be saved. For example, if PAY* is specified as a generic member name, and the system is unable to find a member whose name starts with PAY, the file is not saved. If files specified by the FILEMBR parameter are not saved because members with the specified generic name cannot be found, a diagnostic message is sent, the save operation ends, and an escape message is sent specifying the number of files not saved. If at least one of the files processed for the FILEMBR parameter contains a member with the specified generic name, the diagnostic message is not sent, and the number of files not saved is in the final completion message.

ACCPTH

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it controls only the saving of the access paths. More information on the restoring of saved access paths is in the Backup, Recovery, and Availability topic in the Information Center.

Attention!

If the based-on physical files and the logical files are in different libraries, the access paths are saved.

However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt.

For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

***NO:** Only objects specified on the command are saved. No logical file access paths are saved.

***YES:** The specified physical files and all eligible logical file access paths over them are saved.

Note: Specifying this value does not save the logical files.

SAVFDTA

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

***YES:** The description and contents of the save file are saved. If SAVFDTA(*YES) and STG(*FREE) are specified, storage is freed for the save file.

***NO:** Only the description of a save file is saved.

STG Specifies whether the system storage occupied by the data portion of the specified members (except for save files), modules, programs, service programs, Structured Query Language (SQL) packages, and journal receivers being saved is freed after the save operation is finished. Only the data portion of the object is freed, not the object's description.

***KEEP:** The storage occupied by the data portion of the objects being saved is not freed.

***FREE:** The storage occupied by the data portion of the specified objects is freed as part of the save operation. The storage is freed only after all objects are saved successfully.

Note: To prevent the possible abnormal end of a program, the program being saved must not be running in the system when *FREE is specified.

DTACPR

Specifies whether data compression is performed.

***DEV:** If the tape device has the hardware compression feature installed, processing proceeds as if DTACPR(*YES) is specified. If the compression feature is not installed or if save data is written to a diskette or save file, processing proceeds as if DTACPR(*NO) is specified.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** No data compression or decompression occurs.

***YES:** If the save operation is to tape and the target device has the hardware compression feature, hardware compression is done. If the feature is not present or if the save data is written to optical, diskette, or save file, software compression is done. If the save operation is running while other jobs on the system are active and software compression is used, overall system performance may be affected.

COMPACT

Specifies whether device data compaction is performed.

***DEV:** Device data compaction is performed if the data is saved to tape and all tape devices specified on the DEV parameter support the compaction feature.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** Device data compaction is not performed.

OMITOBJ

Specifies the objects to be excluded from the save operation. Up to 300 objects or generic object values can be specified.

If the OMITOBJ parameter is not specified, all the objects are saved from the specified library.

***NONE:** No objects are excluded from the save operation.

Element 1: Object Name

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

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

***ALL:** Objects in the specified libraries are excluded, depending on the value specified for the object type.

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

object-name: Specify the name of the object that will be excluded from the save command.

Element 2: Object Type

***ALL:** All object types are excluded from the save operation, depending on the value specified for the object name.

object-type: Specify the type of the object to be excluded from the save operation.

» **ASPDEV**

Specifies the name of the auxiliary storage pool (ASP) device to be included in the save operation.

***:** The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

***SYSBAS:** The system ASP and all basic user ASPs are included in the save operation.

***CURASPGRP:** If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

auxiliary-storage-pool-device-name: Specify the name of the independent ASP device to be included in the save operation. «

OUTPUT

Specifies whether a list of information about the saved objects and members is created. The information can be printed with the job's spooled output or directed to a database file.

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

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

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

Note: If OUTPUT(*OUTFILE) is specified, you must specify the database file name on the OUTFILE parameter.

OUTFILE

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

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

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

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

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

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

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

INFTYPE

Specifies the type of information which is printed or directed to the database file.

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

***ERR:** The list contains information about the command, an entry for each library, and an entry for each object that was not successfully saved.

***LIB:** The list contains an entry for each library requested to be saved.

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

OPTFILE

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

'*': The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/':* The system generates an optical file name in the specified directory of the optical volume.

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

USEOPTBLK

Specifies whether or not the optimum block size is used for the save operation.

***YES:** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

If the target release value that is specified is earlier than V3R7M0 then the block size supported by all device types is used.

***NO:** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Examples for SAVOBJ

Example 1: Saving Program and File With Same Name

```
SAVOBJ   OBJ(PETE)  LIB(LIBX)  DEV(DKT01)
```

This command saves the objects named PETE which are located in the LIBX library. If, for example, LIBX contains both a program and a file named PETE, both objects are saved. The storage occupied by the object is not freed because the STG parameter default (*KEEP) was assumed.

Example 2: Freeing System Storage

```
SAVOBJ   OBJ(MSTRPAY PAY*)  LIB(QGPL)  DEV(DKT01)
  STG(*FREE)
```

The object named MSTRPAY, and all the objects whose names start with the characters PAY located in the general purpose library (QGPL), are saved. The objects are copied on diskettes. As part of the save operation, the system storage that was occupied by the data portion of the saved member, module, program, service program, SQL package, and journal receiver objects is freed.

Example 3: Saving File on Diskette

```
SAVOBJ   OBJ(FILEA)  OBJTYPE(*FILE)  LIB(LIBY)
  DEV(DKT01)  VOL(TOM)  CLEAR(*ALL)
```

The file named FILEA in the LIBY library is saved on the diskette. The diskette must be identified by the volume identifier TOM. If the diskette was not cleared, it is cleared automatically before FILEA is saved on it.

Example 4: Saving Objects Supported on Previous Release

```
SAVOBJ   OBJ(PAY*)  LIB(LIB1)  DEV(TAP01)
  TGTRLS(*PRV)
```

This command saves the objects beginning with the characters PAY from the LIB1 library in a format compatible with the previous release of the OS/400 system. Only those objects supported on the previous release are saved.

Example 5: Saving Description and Data of File

```
SAVOBJ   OBJ(SAVEFILE)  LIB(MYLIB)  OBJTYPE(*FILE)
  DEV(DKT01)  SAVFDTA(*YES)
```

This command saves the file named SAVEFILE which is located in the library named MYLIB. Both the description and the data are saved for this save file.

Error messages for SAVOBJ

<u>*ESCAPE Messages</u>

CPFACDB
M36 &1 in library &2 is active. Request not allowed.

CPF3701
&1 objects were saved; &2 objects were not saved.

CPF3702
&1 objects saved; &2 not saved; &9 not included.

CPF3703
&2 &1 in &3 not saved.

CPF3708
Save file &1 in &2 too small.

CPF3709
Tape devices do not support same densities.

CPF3727
Duplicate device &1 specified on device name list.

CPF3728
Device &1 specified with other devices.

CPF3730
Not authorized to &2 &1 in library &3.

CPF3731
Cannot use &2 &1 in library &3.

CPF3733
&2 &1 in &3 previously damaged.

CPF3735
Storage limit exceeded for user profile &1.

CPF3738
Device &1 used for save or restore is damaged.

CPF3747
Object names cannot be specified with more than one library.

CPF3749
Objects from library &2 not saved.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF377D
Save ended because of read error on internal system resource.

CPF377E
Not enough storage for save-while-active request.

CPF377F
Save-while-active request prevented by pending record changes.

CPF3770
No objects saved or restored for library &1.

CPF3771
&1 objects saved from &3. &2 not saved.

CPF3774
Not all objects saved from library &3.

CPF3778
Not all objects saved from all libraries.

CPF378A
Message queue not available.

CPF378C
SAVACTMSGQ(*WRKSTN) not valid for batch job.

CPF378E
Library &1 not saved.

CPF3781
Library &1 not found.

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

CPF3789
Only one library allowed with specified parameters.

CPF379E
Not enough storage available to save library &1.

CPF3790
No available space on diskette.

CPF3793
Machine storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3797
Objects from library &3 not saved. Save limit exceeded.

CPF380B
Save cannot be completed at this time.

CPF3812
Save file &1 in &2 in use.

CPF3815
Save file &1 in &2 too small for save operation.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3867
Contents of FILEMBR parameter not correct.

CPF3868
FILEMBR specified but OBJTYPE must be *ALL or *FILE.

CPF3871
No objects saved or restored; &3 objects not included.

CPF3892
&2 &1 in &3 not saved.

CPF3894
Cancel reply received for message &1.

CPF5729
Not able to allocate object &1.

CPF9809
Library &1 cannot be accessed.

CPF9812
File &1 in library &2 not found.

» **CPF9814**
Device &1 not found. «

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

» **CPF9833**
*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group. «

» **CPFB8ED**
Device description &1 not correct for operation. «

OPT1332
Optical volume not found in device &1.

*STATUS Messages

CPF3770
No objects saved or restored for library &1.

CPF3771
&1 objects saved from &3. &2 not saved.

CPF3871
No objects saved or restored; &3 objects not included.

SAVOBJLBRM (Save Object List Using BRM) Command Description

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

SAVOBJLBRM Command syntax diagram

Purpose

The Save Object List using BRM (SAVOBJLBRM) saves objects that are specified in a BRMS object list. It does not save the list itself.

» **Restrictions:**

- You must have authority to the SAVOBJ or SAVCHGOBJ commands to use this command.
- You must have *USE authority to any auxiliary storage pool device specified for the ASPDEV parameter.
- When saving to a save file, only one library can be specified.
  «
- No object being saved can be changed by a job that is running at the time the save operation occurs unless save-while-active is used.
- This command should not be used by control group *EXIT item processing as results will be unpredictable.

Example for SAVOBJLBRM

Example 1: Saving All Objects in a List

```
SAVOBJLBRM OBJL(LISTO)
DEV(*MEDCLS *MEDCLS *MEDCLS)
MEDPCY(FULL)
```

In the example, all objects in an object list (LISTO) will be saved to any three devices that support the density specified in the media class in the media policy FULL.

Error messages for SAVOBJLBRM

None »

SAVBRM (Save Object Using BRM) Command Description

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

SAVBRM Command syntax diagram

Purpose

The Save Object using BRM (SAVBRM) command saves a copy of one or more objects that can be used in the integrated file system.

For more information about the integrated file system commands, see the Integrated File System Introduction topic in the Information Center.

Restrictions:

1. For detailed restrictions on using this command to save objects by using name patterns in the root directory, to save objects in libraries, or to save documents library objects, see the Backup and Recovery book.
2. The SAVBRM command does not support optical files.
3. You cannot save QSYS and QDLS file systems using this command.

You can save data to an ADSM device by using this command. To save data to an ADSM device, you must meet the following conditions:

- You must specify the special value *ADSM in both the MEDCLS and MOVPCY parameters.
- You must specify a device that supports ADSM backups. You can only specify one such device in the list of devices. You can specify the *MEDCLS special value in the DEV parameter, which can select an ADSM type of device.
- You can save only user data. BRMS does not save *IBM type libraries to ADSM media. BRMS does not save IBM-supplied libraries that are considered user data such as QGPL, QUSRBRM, or QUSRSYS libraries to ADSM media.
- When you perform a save to an ADSM device, you must specify *NO for the SAVF parameter. If you specify SAVF(*MEDPCY), you must specify *NO for the *Save to save file* field in the referred to media policy.
- Designation of any release other than *CURRENT in the TGTRLS parameter is ignored and changed to *CURRENT.

Example for SAVBRM

Example 1: Saving Objects

```
SAVBRM DEV(*MEDCLS)
MEDPCY(*SYSPCY)
```

In this example all integrated file system objects are included in the save operation using the device specified in the system policy.

Error messages for SAVBRM

No error messages. «

SAVOBJBRM (Save Object using BRM) Command Description

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

SAVOBJBRM Command syntax diagram

Purpose

The Save Object using BRM (SAVOBJBRM) command saves individual objects or a group of objects that are in the same library. The system saves the specified objects by writing a copy of each object on the volume, or in a save file. The objects are not affected in the system unless the command specifies that the storage should be freed.

Restrictions:

- » You must have authority to the SAVOBJ or SAVCHGOBJ commands to use this command.
- You must have *USE authority to any auxiliary storage pool device specified for the ASPDEV parameter.
- When saving to a save file, only one library can be specified.
  «
- No object being saved can be changed by a job that is running at the time the save operation occurs unless save-while-active is used.
- This command should not be used by control group *EXIT item processing as results will be unpredictable.

Note: Do not precede an entry with an asterisk unless that entry is a "special value" that is shown (on the display itself or in the help information) with an asterisk.

Examples for SAVOBJBRM

Example 1: Saving Objects of a Particular Type

```
SAVOBJBRM LIB(MBALIB) OBJ(AP* GLMST) DEV(TAP09)
OBJTYPE(*FILE) MEDPCY(FULL)
```

In this example, selected file objects whose names begin with AP, or the object named GLMST, are saved from library MBALIB using TAP09.

Example 2: Saving Generic Objects from a Library

```
SAVOBJBRM LIB(MBALIB) OBJ(DIST*) DEV(TAP09)
MEDPCY(FULL)
```

In this example all objects are saved that begin with DIST from library MBALIB to device TAP09.

Error messages for SAVOBJBRM

None

SAVSAVFDTA (Save Save File Data) Command Description

SAVSAVFDTA Command syntax diagram

Purpose

The Save Save File Data (SAVSAVFDTA) command saves the contents of a save file to tape, optical, or diskette media. This command saves the save data in the save file to the device in a way that allows the user to restore objects directly from the device.

A save file containing data created by the Save Security Data (SAVSECDTA) or Save Configuration (SAVCFG) command can only be saved to tape or optical. A save file containing data created by the Save Licensed Program (SAVLICPGM) command cannot be saved by this command.

The information written on media by this command is similar to the data that was previously written to the save file by the save command that originally created the save file data. This includes the object descriptions, and object contents that existed when the original save operation was done.

This command uses only the save file and device description objects; it does not refer to or modify the description or contents of the objects included in the file save data. Thus, objects included in the save file are not locked during the running of this command, and the save history information (date, place, and time when each object was last saved) is not updated by this command for each object in the save file.

The description of the save file is not included in the save operation (unless it was included with the objects that were saved to create the save data in the file). In addition, this command does not update the save history information for the save file object, so the last save operation date, time, and place always identify the last save operation of the save file object description, not its contents.

Note: This command ignores all file overrides currently in effect for the job, except for the output file.

Restrictions:

1. The user of this command must have use authority for the save file and use authority for the tape, optical, or diskette device description.
2. All diskettes to be used to perform the save operation must be initialized in the save/restore format. If a tape is used, it must have a standard label. Also, the save file cannot be in use by a job running at the time the save operation occurs.
3. Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

Required Parameters

SAVF Specifies the qualified name of the save file whose contents are saved. If no library qualifier is specified, *LIBL is used to find the file.

Note: The save file must contain the data made by the run of a previous save command or an error message is sent, and its contents are not saved to diskette or tape.

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

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

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

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

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

DEV Specifies the names of the devices used for the save operation. Each device name must already be known on the system by a device description.

diskette-device-name: Specify the name of the diskette device used to save the data.

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

tape-media-library-device-name: Specify the name of the tape media library device used to save the data.

tape-device-name: Specify the name of one or more tape devices that are used for the save operation. If more than one tape device is used, specify the names of the devices in the order they are used. Up to four device names can be specified. When more than one tape volume is used, it may be advantageous to use more than one tape device so that tape volumes can be rewound and/or unloaded while another tape device is being processed.

Optional Parameters

VOL Specifies the volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in Commonly used parameters.

***MOUNTED:** The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

SEQNBR

Specifies, when tape is used, the sequence number on the tape where the save operation begins.

***END:** The system saves the data starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

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

ENDOPT

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

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

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

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

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

CLEAR

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. Clearing active data removes all files from the volume, starting at the specified sequence number for tape. Replacing active data on optical media replaces only the optical files created by this operation.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. Clearing a diskette does not initialize it. You should initialize diskettes to a save and restore format before the save command is issued by using the Initialize Diskette (INZDKT) command and specifying FMT(*SAVRST).
4. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

***NONE:** None of the media is automatically cleared. If the save operation encounters active data on a tape, diskette, or save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL:** All of the media is automatically cleared.

If tapes are used and a sequence number is specified on the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER:** All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape or diskette, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***REPLACE:** Active data on the media is automatically replaced. Optical volumes are not initialized. Tapes, diskettes, and save files are automatically cleared in the same way as the *ALL value.

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

PERM: The file written on diskette or tape is protected permanently.

expiration-date: Specify the date on which protection of the file on diskette or tape ends. The date must be specified in job date format.

COMPACT

Specifies whether device data compaction is performed.

***DEV:** Device data compaction is performed if the data is saved to tape and all tape devices specified on the DEV parameter support the compaction feature.

***NO:** Device data compaction is not performed.

OUTPUT

Specifies whether a list with information about the saved contents of a save file is created. The information can be printed with the job's spooled output or directed to a database file.

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

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

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

Note: If OUTPUT(*OUTFILE) is specified, you must specify the database file name on the OUTFILE parameter.

OUTFILE

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

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

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

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

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

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

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

OPTFILE

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

'*': The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/':* The system generates an optical file name in the specified directory of the optical volume.

USEOPTBLK

Specifies whether or not the optimum block size is used for the save operation.

***YES:** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

If the target release value that is specified is earlier than V3R7M0 then the block size supported by all device types is used.

***NO:** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Examples for SAVSAVFDTA

Example 1: Clearing All Diskettes and Tapes Encountered

```
SAVSAVFDTA  SAVF(ONLINE)  DEV(TAP01)
  SEQNBR(1)  CLEAR(*ALL)
```

This command saves the contents of save file ONLINE to the first file on the tape volume on device TAP01. Files that have not ended on either the first tape volume or on subsequent volumes are overwritten without an inquiry message because CLEAR(*ALL) is specified.

Example 2: Restoring Library from Tape

If, in the last save command to save library USRLIB, the save file was:

```
SAVLIB  USRLIB  SAVF(ONLINE)
```

Then the following command must be specified to restore the library from tape:

```
RSTLIB  USRLIB DEV(TAP01)  VOL(*MOUNTED)
```

Error messages for SAVSAVFDTA

***ESCAPE Messages**

CPF3707
Save file &1 in &2 contains no data.

CPF3709
Tape devices do not support same densities.

CPF3727
Duplicate device &1 specified on device name list.

CPF3728
Device &1 specified with other devices.

CPF3733
&2 &1 in &3 previously damaged.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

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

CPF3793
Machine storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

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

CPF3812
Save file &1 in &2 in use.

CPF3814
No objects from save file &1 in &2 saved to media.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF5729
Not able to allocate object &1.

CPF9812
File &1 in library &2 not found.

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

CPF9825
Not authorized to device &1.

SAVSAVFBRM (Save Save Files Using BRM) Command Description

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

SAVSAVFBRM Command syntax diagram

Purpose

The Save Save Files using BRM (SAVSAVFBRM) command saves save files created during BRMS processing to tape. Various parameters allow you to select which save files that you want to copy to tape. For instance, you can select a range of save file creation dates to include in the save files that you want to copy to tape with further qualification of control group and media class. You can schedule the SAVSAVFBRM command through the system job scheduler, through another job scheduler or on demand. Expired save files that are not waiting to be copied to tape are deleted when the SAVSAVFBRM command processes.

Note: You must have authority to the SAVSAVFDTA command to use this command.

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

Note: Data in a save file created with device *NONE will never be copied by BRMS to tape. The save file is intended for online access only.

Example for SAVSAVFBRM

Example 1: Saving Save Files to Removable Media

```
SAVSAVFBRM CTLGRP(BACK*) MEDCLS(QIC1000)
BEGIN(*AVAIL '1/1/99') END(*AVAIL *CURRENT)
```

In the example, all backup control groups whose name begins with the BACK that created save files with media class QIC1000 are saved to tape. The device to which the save files are to be saved is specified in the save file itself and is thus not specified in the command. The save files that are to be included are save files that were created between 1/1/99 and the current date. Creation time does not matter. Expired save files that are not waiting to be copied to tape are deleted when the command processes.

Error messages for SAVSAVFBRM

None

SAVSECDTA (Save Security Data) Command Description

SAVSECDTA Command syntax diagram

Purpose

The Save Security Data (SAVSECDTA) command saves all security information without requiring a system in a restricted state.

SAVSECDTA saves the same security information that is saved when a SAVSYS command is run including:

- User Profiles
- Authorization Lists
- Authority Holders

Information saved with the SAVSYS command or SAVSECDTA command can be restored with the RSTUSRPRF and RSTAUT commands.

Restrictions:

1. Users of the SAVSECDTA command must have *SAVSYS special authority.
2. Changes made to user profiles while the SAVSECDTA command is being run may not be reflected in the media, depending on when the changes occurred in relation to the save operation.
3. Concurrent running of other SAVSECDTA commands is not allowed.
4. If PRECHK(*YES) is specified and a security object cannot be saved, the save operation ends.
5. Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

Required Parameter

DEV Specifies the name of the device on which the security data is saved. The device name must already be known on the system by a device description.

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

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

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

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

Optional Parameters

VOL Specifies the volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in Commonly used parameters.

***MOUNTED:** The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are placed in the device and used to save the system data. A maximum of 75 volume identifiers can be specified.

SEQNBR

Specifies, only when a tape is used, which sequence number is used as the starting point for the save operation.

***END:** The system saves the file starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

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

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

***PERM:** The file is protected permanently.

expiration-date: Specify the date when the file protection ends.

ENDOPT

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

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

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

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

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

SAVF Specifies the qualified name of the save file used to contain the save data. The save file must be empty or CLEAR(*ALL) must be specified.

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

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

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

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

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

CLEAR

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. Clearing active data removes all files from the volume, starting at the specified sequence number for tape. Replacing active data on optical media replaces only the optical files created by this operation.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. Clearing a diskette does not initialize it. You should initialize diskettes to a save and restore format before the save command is issued by using the Initialize Diskette (INZDKT) command and specifying FMT(*SAVRST).
4. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

***NONE:** None of the media is automatically cleared. If the save operation encounters active data on a tape, diskette, or save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL:** All of the media is automatically cleared.

If tapes are used and a sequence number is specified on the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER:** All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape or diskette, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE:** Active data on the media is automatically replaced. Optical volumes are not initialized. Tapes, diskettes, and save files are automatically cleared in the same way as the *ALL value.

PRECHK

Specifies whether the save operation ends if all security objects specified do not satisfy the following conditions of the save operation:

- The objects exist
- They were not previously found to be damaged
- They are not locked by another job
- The requester of the save operation has authority to save the objects

***NO:** The save operation continues, saving only security objects that can be saved.

***YES:** The save operation ends before any data is written if, after all security objects are checked, one or more objects cannot be saved.

DTACPR

Specifies whether data compression is performed.

***DEV:** If the tape device has the hardware compression feature installed, processing proceeds as if DTACPR(*YES) is specified. If the compression feature is not installed or if save data is written to a diskette or save file, processing proceeds as if DTACPR(*NO) is specified.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** No data compression occurs.

***YES:** Hardware compression is performed if the save operation is to tape and the target device has the hardware compression feature. If the feature is not present or if the save data is written to optical or a save file, software compression is performed. Running the save operation while other jobs on the system are active and software compression is used may affect the overall system performance.

COMPACT

Specifies whether device data compaction is performed.

***DEV:** Device data compaction is performed if the data is saved to tape and all tape devices specified on the DEV parameter support the compaction feature.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** Device data compaction is not performed.

OUTPUT

Specifies whether a list with information about the saved security objects is created. The information can be printed with the job's spooled output or directed to a database file.

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

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

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

Note: If OUTPUT(*OUTFILE) is specified, you must specify the database file name on the OUTFILE parameter.

OUTFILE

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

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

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

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

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

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

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

OPTFILE

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

'*': The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name/':* The system generates an optical file name in the specified directory of the optical volume.

USEOPTBLK

Specifies whether or not the optimum block size is used for the save operation.

***YES:** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

If the target release value that is specified is earlier than V3R7M0 then the block size supported by all device types is used.

***NO:** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

ASPDEV

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

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

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

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

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

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

Examples for SAVSECDTA

Example 1: Automatically Clearing Uncleared Tapes

```
SAVSECDTA  DEV(TAP01)  CLEAR(*ALL)
```

This command saves the security information, including user profiles, authorization lists, authority holders. They are saved on the TAP01 tape drive. CLEAR(*ALL) automatically clears all uncleared tapes when they are encountered.

Example 2: Sending Message When Storage Capacity Exceeded

```
SAVSECDTA  DEV(TAP01)  VOL(ABC)
```

This command saves the security information on the TAP01 tape drive, starting on the tape volume labeled ABC. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be put on the TAP01 tape drive is shown to the operator.

Error messages for SAVSECDTA

<u>*ESCAPE Messages</u>

CPF370A
Not all security objects saved to save file &3.

CPF3709
Tape devices do not support same densities.

CPF3727
Duplicate device &1 specified on device name list.

CPF3728
Device &1 specified with other devices.

CPF3731
Cannot use &2 &1 in library &3.

CPF3733
&2 &1 in &3 previously damaged.

CPF3735
Storage limit exceeded for user profile &1.

CPF3737
Save and restore data area &1 not found.

CPF3738
Device &1 used for save or restore is damaged.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

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

CPF3793
Machine storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3812
Save file &1 in &2 in use.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3893
Not all security objects saved.

CPF3894
Cancel reply received for message &1.

CPF5729
Not able to allocate object &1.

CPF9809
Library &1 cannot be accessed.

CPF9812
File &1 in library &2 not found.

» **CPF9814**
Device &1 not found. «

» **CPF9833**
*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group. «

CPF9845
Error occurred while opening file &1.

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

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

CPF9850
Override of printer file &1 not allowed.

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

CPF9860
Error occurred during output file processing.

» **CPFB8ED**
Device description &1 not correct for operation. «

SAVSTG (Save Storage) Command Description

SAVSTG Command syntax diagram

Purpose

The Save Storage (SAVSTG) command saves a copy of the licensed internal code and the contents of auxiliary storage (except unused space and temporary objects) to tape. This function is intended for disaster recovery backup. Individual libraries or objects cannot be restored from a save storage tape.

This command issues the PWRDWNSYS (Power Down System) command with OPTION(*IMMED) and RESTART(*YES) specified. While the system is powered down, a dedicated service tool (DST) that saves all system storage is called. At that time, a standard labeled tape volume must be placed in the tape device. Additional volumes are requested as needed. Hardware data compression is used if it is supported by the tape device. After the save system storage function is complete, an initial program load (IPL) takes place.

During the IPL after a save storage operation, a completion message is sent to the system operator message queue. The save history information for the data area QSAVSTG in library QSYS is updated with the date and time when the system storage data was saved. To show the information in this data area, use the Display Object Description (DSPOBJD) command with DETAIL(*FULL).

The restore storage operation is done using the appropriate option on the DST menu. During the IPL, after a restore storage operation, a completion message is sent to the system operator message queue, and the last restore date and time history information in the QSAVSTG data area is updated with the current date and time. In addition, the data portion of the QSAVSTG data area is updated with the date of the save storage tape used in the restore system storage operation.

Note: Because media errors cause the save operation to start over from the last tape volume, use of this command is recommended for smaller systems only.

Restrictions:

1. To use this command, you must have *SAVSYS special authority.
2. The system cannot be running any other jobs; (ENDSBS SBS (*ALL) or ENDSYS must be issued).
3. Tapes created using this command that will be used for installation should be initialized with a density that is supported by the current alternative IPL tape unit. If this is not done, the current IPL tape will have to be changed to a tape device that supports the density of the created SAVSTG tapes before installation begins.
4. Tapes created using the SAVSTG command should not be used for automatic installation.

Required Parameter

DEV Specifies the name of the tape device used for the save storage operation. Specify the name of the tape device.

Optional Parameters

EXPDATE
Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

***PERM:** The tape files are permanently protected.

expiration-date: Specify the date when the tape files end. The date must be specified in the job date format. If separators, specified by the system value QDATSEP, are used, the value must be enclosed in apostrophes.

CLEAR

Specifies whether uncleared tapes encountered during the save operation are automatically cleared. An uncleared tape is one containing a file with an expiration date later than the date of the save operation (including files protected permanently with EXPDATE(*PERM)).

Note: This parameter does not control initializing the tapes. The tapes used to perform the save operation should be initialized to a standard label format before the save command is issued. You can use the Initialize Tape (INZTAP) command and specify a value on the NEWVOL parameter to initialize a tape to a standard label format. If a tape volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the tape volume.

***NONE:** None of the uncleared tapes encountered during the save operation are automatically cleared. If the save operation cannot continue because an uncleared tape is encountered, an inquiry message is sent to the operator allowing the save operation to end, specifying that the currently selected tape should be cleared so the operation can continue. All tapes used to perform the save operations should be initialized before the save command is issued.

***ALL:** All uncleared tapes encountered during the save operation are automatically cleared.

Examples for SAVSTG

Example 1: Specifying Expiration Date

```
SAVSTG  DEV(TAP01)  EXPDATE(122290)  CLEAR(*ALL)
```

This command saves the system storage on the tape put on the TAP01 tape drive. Each uncleared tape is cleared automatically. The tape files written are protected and cannot be overwritten until December 22, 1990.

Example 2: Saving System Storage

```
SAVSTG  DEV(TAP02)
```

The system storage is saved on tape drive TAP02. CLEAR was not specified, so uncleared tapes encountered during the save operation cause an inquiry message to be sent to the operator, who either ends the save operation or specifies that the currently selected tape be cleared so the operation can continue. Because EXPDATE also was not specified, the tape files being written are protected permanently.

Error messages for SAVSTG

*ESCAPE Messages

CPF2206

User needs authority to do requested function on object.

CPF376A

System must not be in manual mode.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF3785
Not all subsystems ended.

SAVSYS (Save System) Command Description

SAVSYS Command syntax diagram

Purpose

The Save System (SAVSYS) command saves a copy of the licensed internal code and the QSYS library in a format compatible with the installation of the iSeries 400. It does not save objects from any other library. In addition, it saves the security and configuration objects that can also be saved using the SAVSECDTA and SAVCFG commands.

To save the system data on offline storage, the system writes a copy of the objects onto the media. The libraries and objects are not affected on the system. This command cannot be used to free any space occupied by these objects. The history information for the data area QSAVUSRPRF in QSYS is updated with the date, time, and place where user profiles are saved. The history information for the data area QSAVSYS in QSYS is updated with the date, time, and place where the system is saved. The history information for the data area QSAVCFG in QSYS is updated with the date, time, and place where configuration objects are saved. The history information is not updated for the individual objects. To display the information in these data areas, specify the Display Object Description (DSPOBJD) command, and specify DETAIL(*FULL). Save the information from the display of QSAVUSRPRF for the location where the user profiles are saved.

When using this command, it is important to use the device on the system that is defined as the initial program load (IPL) device. The IPL device was defined by the service representative when the system was installed. If an IPL device is not used when using this command, then the system cannot be restored using the SAVSYS media (if densities or media types are incompatible).

Restrictions:

1. You must have *SAVSYS special authority to use this command.
2. All subsystems must be inactive before the SAVSYS command can be specified. The End System (ENDSYS) or End Subsystem (ENDSBS) command can be used to make the subsystems inactive. You must have job control (*JOBCTL) authority to use the ENDSYS or the ENDSBS command.
3. Tapes created using this command that will be used for installation should be initialized with a density that is supported by the current IPL tape unit. If this is not done, the current IPL tape will have to be changed to a tape device that supports the density of the created SAVSYS tapes before installation begins.
4. Tapes created using the SAVSYS command should not be used for automatic installation.
5. Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

Required Parameter

DEV
Specifies the names of the tape devices, optical device, or the name of the media library device used for the save system operation. Each device name must be known on the system by a device description. If multiple devices are specified, they must have compatible media formats.

If more than one tape device is used, specify the names of the devices in the order in which they are used. Up to four tape device names and one tape media library device name can be specified.

When more than one tape volume is used, more than one tape device may be used so that one tape volume can be rewound and unloaded while another tape device is being processed.

Optional Parameters

VOL Specifies the volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in Commonly used parameters.

***MOUNTED:** The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

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

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

***PERM:** The files are permanently protected.

expiration-date: Specify the date when protection of the files ends.

ENDOPT

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

Note: For optical devices, *UNLOAD is the only special value supported, *REWIND and *LEAVE will be ignored.

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

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

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

OMIT Specifies what data to omit from the save system operation.

***NONE:** All of the security data and configuration data are included in the save system operation.

***CFG:** All of the configuration data is omitted from the save system operation. You can use the SAVCFG (Save Configuration) command to save just the configuration object information.

***SECDTA:** All of the security data is omitted from the save system operation. You can use the SAVSECDTA (Save Security Data) command to save just the system security data.

OUTPUT

Specifies whether a list with information about the saved objects is created. The information can be printed with the job's spooled output or directed to a database file. One label entry is created for each of the files on the media.

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

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

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

Note: If OUTPUT(*OUTFILE) is specified, you must specify the database file name on the OUTFILE parameter.

OUTFILE

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

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

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

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

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

database-file-name: Specify the name of the database file to which the output of the command is directed when OUTPUT(*OUTFILE) is specified.

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

CLEAR

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. Clearing active data removes all files from the volume. Replacing active data on optical media initializes the first volume, and replaces only the optical files created by this operation on any volumes after the first volume.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.
2. Clearing an optical volume does initialize it. The first optical volume is always initialized.

3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

***NONE:** None of the media is automatically cleared. If the save operation encounters active data on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters active data after the first volume, an inquiry message is sent, allowing the operator to either end the save operation, clear the tape, or replace the optical file.

***ALL:** All of the media is automatically cleared.

***AFTER:** All media after the first volume is automatically cleared. If the save operation encounters active data on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media.

***REPLACE:** Active data on the media is automatically replaced. The first optical volume is initialized. Other optical volumes are not initialized. Tapes are automatically cleared in the same way as the *ALL value.

DTACPR

Specifies whether data compression is performed.

***DEV:** If the tape device has the hardware compression feature installed, processing proceeds as if DTACPR(*YES) is specified. If the compression feature is not installed or if save data is written to a diskette or save file, processing proceeds as if DTACPR(*NO) is specified.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** No data compression is performed.

***YES:** If the target device has the hardware compression feature, hardware compression is performed. If the feature is not present, software compression is performed.

COMPACT

Specifies whether device data compaction is performed.

***DEV:** Device data compaction is performed if the data is saved to tape and the target device supports the compaction feature.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** Device data compaction is not performed.

USEOPTBLK

Specifies whether or not the optimum block size is used for the save operation.

***YES:** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

If the target release value that is specified is earlier than V3R7M0 then the block size supported by all device types is used.

***NO:** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Note: Not all files will be saved with optimum block size.

» **ASPDEV**

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

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

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

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

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

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

Examples for SAVSYS

Example 1: Tapes Cleared Automatically

```
SAVSYS  DEV(TAP01)  CLEAR(*ALL)
```

This command saves licensed internal code, system objects, all user profiles (including private authority for objects), and all line, controller, and device descriptions. They are saved on the tape put on the TAP01 tape drive. Each uncleared tape is automatically cleared when it is encountered, and the save operation continues without operator intervention.

Example 2: Operating Receives Message of Exceeded Storage Capacity

```
SAVSYS  DEV(TAP01)  VOL(ABCDE)
```

The system data is saved on the TAP01 tape drive, starting on the tape volume labeled ABCDE. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be put on the TAP01 tape drive is shown to the operator.

Example 3: Saving Data on Two Tape Drives in Alternating Order

```
SAVSYS  DEV(TAP01 TAP02)
```

The system data is saved on tape drives TAP01 and TAP02 in alternating order. If the save operation exceeds the storage capacity of two tapes, a message requesting that another volume be put on TAP01 is shown to the operator. The tapes are rewound at the completion of the save system operation.

Error messages for SAVSYS

***ESCAPE Messages**

CPF2206
User needs authority to do requested function on object.

CPF3703
&2 &1 in &3 not saved.

CPF3709
Tape devices do not support same densities.

CPF372B
Not all objects were saved.

CPF3727
Duplicate device &1 specified on device name list.

CPF3728
Device &1 specified with other devices.

CPF3733
&2 &1 in &3 previously damaged.

CPF3735
Storage limit exceeded for user profile &1.

CPF3738
Device &1 used for save or restore is damaged.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF3772
SAVSYS completed. One or more objects not saved.

CPF3785
Not all subsystems ended.

CPF3793
Machine storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3797
Objects from library &3 not saved. Save limit exceeded.

CPF3798
Installation &2 &1 in &3 not found.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3873
Licensed program &1 option &2 release &4 not saved.

CPF3894
Cancel reply received for message &1.

» **CPF9814**
Device &1 not found. «

» **CPF9833**
*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group. «

CPF9845
Error occurred while opening file &1.

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

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

CPF9850
Override of printer file &1 not allowed.

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

CPF9860
Error occurred during output file processing.

» **CPFB8ED**
Device description &1 not correct for operation. «

SAVSYSBRM (Save System using BRM) Command Description

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

SAVSYSBRM Command syntax diagram

Purpose

The Save System using BRM (SAVSYSBRM) command saves a copy of the Licensed Internal Code and the QSYS library in a format compatible with the installation process. It does not save objects from any other library. The SAVSYSBRM command saves all object types shown on the OBJTYPE parameter in the Save Object (SAVOBJBRM) command. In addition, it saves security and configuration objects.

Restrictions:

- You must have authority to the SAVSYS command to use this command.
- You must have *SAVSYS special authority to use this command.
- « You must have *USE authority to any auxiliary storage pool device specified for the ASPDEV parameter, or to all online auxiliary storage pool devices if *ALLAVL is specified as the value for the ASPDEV parameter.
  «
- All subsystems will be ended when using the SAVSYSBRM command. You must have job control (*JOBCTL) authority to use the ENDSYS or the ENDSBS command.
- Volumes created using this command that will be used for installation should be initialized with a media class with a density that is supported by the current alternate IPL device. If this is not done, the current IPL volume will have to be changed to a device that supports the density of the created SAVSYSBRM volumes before installation begins.

- Volumes created using the SAVSYSBRM command cannot be used for automatic installation.
- The media class associated with the device must specify Shared media *NO.

Note: Do not precede an entry with an asterisk unless that entry is a “special value” that is shown (on the display itself or in the help information) with an asterisk.

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

Example for SAVSYSBRM

Example 1: Saving the System

```
SAVSYSBRM DEV(TAP09) MEDPCY(SAVSYS)
```

This command saves a copy of the Licensed Internal Code and the QSYS library. In this example you are saving to device TAP09 using a media policy called SAVSYS.

Error messages for SAVSYSBRM

None

SAVS36F (Save System/36 File) Command Description

SAVS36F Command syntax diagram

Purpose

The Save System/36 File (SAVS36F) command saves the following:

- A copy of a single database physical or logical file to diskette, tape or another database physical file in the same format as if a System/36 had saved the file.
- A copy of multiple database physical or logical files to diskette or tape in the same format as if a System/36 had saved the files as a **Save All Set**. The following groups of files can be saved together with one operation:
  - All files in a library
  - All files in a specific **file group**
  - All files that are not part of a file group
  - All files that begin with a specified set of characters

A **Save All Set** is a group (set) of files that share the same group (set) name and that are saved (copied) to diskette or tape with one operation. The set of files can be restored (copied back from diskette or tape) with a single operation by referring to the set name (see the Restore System/36 Files (RSTS36F) command).

File groups are defined by file names that contain a period. The characters preceding the period identify the file group, and the characters following the period identify the file within the group. As with file names within the System/36 environment, the maximum number of characters is eight, including the period. Files with names that do not contain a period are not part of a file group. The following examples show the names of files within a file group.

```
PAYROL.A
PAYROL.B    Files in File Group PAYROL
PAYROL.C

A.ACCTS
A.INV
A.PROL      Files in File Group A
```

```
A.B.GO
A.B.INV

A.B.GO
A.B.INV     Files in File Group A.B
```

The saved files can be restored to the following systems:

- System/36 (RESTORE procedure or $COPY utility)
- iSeries 400 (Restore System/36 File (RSTS36F) command)

The SAVS36F command is intended for exchanging files with a System/36. For creating a backup version of a file, the iSeries 400 save commands (for example, Save Object (SAVOBJ) or Save Changed Object (SAVCHGOBJ)) should be used.

Restrictions:

1. The following authorities are required (normally only applies when running on a system using resource security):
   - *USE authority for this command.
   - *USE authority for the file or group of files specified in the FROMFILE parameter.
   - *USE authority for the library specified in the FROMLIB parameter.
   - *CHANGE authority to the file specified on the PHYFILE parameter if saving to an existing physical file.
   - *USE authority for the library specified in the PHYFILE parameter if saving to a physical file.
   - *CHANGE authority for the library specified in the PHYFILE parameter if saving to a physical file and the file does not exist.
   - *USE authority for the diskette device description object, *USE authority for device file QSYSDKT, in library QSYS if saving to diskette.
   - *USE authority for the tape device description object, *USE authority for device file QSYSTAP, in library QSYS if saving to tape.
   - *USE authority for the based-on physical file if saving a logical file.
2. All diskettes that are used for the save operation should be initialized using the INZDKT CL command or the equivalent System/36 environment function (INIT operator control language (OCL) procedure or $INIT SSP utility). For a two-sided diskette, use a sector size of 256 or 1024. For a one-sided diskette, use a sector size of 128 or 512. If tape is used, each tape volume used should have been initialized with standard labels using the INZTAP CL command or the equivalent System/36 environment function (TAPEINIT OCL procedure or $TINIT SSP utility). Use a density of 1600 bits per inch when initializing the tape.

Note: If the tape or diskette has not been initialized as stated above, the System/36 will not be able to process the media.

3. Object-level and record-level functions, other than read operations, should not be attempted for a file being saved by SAVS36F. Concurrent activity against the file (for example, moving the file or adding or removing records) can cause:
   - For a save operation of a single file (FROMFILE(file-name)), the save operation will end with escape message CPF9826 because the file cannot be allocated.
   - For a save operation of multiple files (FROMFILE(*ALL or generic*-file-name)), the save function sends an inquiry message CPA2C6A because the file cannot be allocated. The message allows an ignore, retry and cancel response. The ignore response bypasses this file and attempts to save the next file selected.
4. When saving a single file to diskette, the diskette cannot already contain an active file with the same label and creation date as the new file to be created.

5. When saving multiple files to diskette, the diskette used for the save cannot contain any active files.
6. Not all physical and logical files can be saved with the SAVS36F command.
   - Only logical files created under the System/36 environment (for example, through the BLDINDEX OCL procedure) or through a DDM request from a System/36 system can be saved. These files are saved as System/36 alternative index files.
   - All physical files created under the System/36 environment (for example, through the BLDFILE OCL procedure) or through a DDM request from a System/36 system are saved using information stored within the iSeries 400 file description. These files are saved as System/36 sequential, direct, or indexed physical files.
   - Any physical files created by iSeries 400 commands or utilities can be saved as long as the record length is not greater than 4096. These files are saved as System/36 sequential files.
7. To generate a save format which can be processed by the System/36 RESTORE procedure, the following information is not saved:
   - If saving a logical file, only the description of the file is saved. The index (or access path) is not saved.
   - If saving an indexed (keyed) physical file, the data is saved but the index is not. The index will be rebuilt after the file is restored.
8. The following restrictions apply to naming standards:
   - When saving a single file, the specified name (FROMFILE parameter) must meet naming standards. If not, message CPF0001 is sent when the SAVS36F command is processed.
   - If a file name is found during a save operation of multiple files (FROMFILE(*ALL or generic*-file-name)) that does not meet the System/36 naming standards, diagnostic message CPF2C0E is sent and the file is not saved.
9. Multiple files (FROMFILE(*ALL) or FROMFILE(generic*-name)) cannot be saved to a physical file.

Required Parameters

FROMFILE

Specifies the name of one or more files to save. The files being saved must be in the library specified in the FROMLIB parameter.

***ALL:** All files in the specified library are saved. The GROUP parameter is used to further describe which files are saved.

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

file-name: Specify the name of a single file to save.

FROMLIB

Specifies which library contains the database files to be saved.

DEV Specifies the names of the devices used for the save operation. Each device name other than *PHYFILE must already be known on the system by a device description.

***PHYFILE:** Specifies a database physical file is to receive the copied file. The qualified name of the physical files must be specified on the PHYFILE parameter.

*PHYFILE is not valid when saving multiple files by specifying *ALL or generic*-file-name on the FROMFILE parameter.

If the physical file does not exist, it is created as a non-keyed, program-described file with a record length of 256. If a file already exists by this name, it is used as long as it is a non-keyed physical file with a record length of 256.

The copied records are put in the first member of the physical file. If the file has no members, a member is created using the name syntax 'Myymmdd' and the system date (for example, if the SAVS36F was run on June 28, 1987, the member name would be M870628).

device-name: Specify the name of the diskette device, the media library device, or the names of tape devices that are used for the operation. If more than one tape device is used, specify the names of the devices in the order in which they are used. When more than one tape volume is used, using more than one tape device permits one tape volume to be rewound or unloaded while another tape device processes the next tape volume. More information on device names is in the APPC, APPN, and HPR topic in the Information Center.

Optional Parameters

GROUP

Specify which file groups are to be saved. This parameter is only valid if *ALL is specified for the FROMFILE parameter.

***ALL:** All files are saved.

***NONE:** Only files that do not belong to a file group are saved.

group-name: Specify the name of a file group to be saved. Do not specify either the period (.), which indicates a file name, or the file name itself. For example, to save files belonging to the file group that includes PAYROL.A, PAYROL.B, and PAYROL.C, enter PAYROL for this parameter. Files that do not belong to the specified file group are not saved. A group-name can be up to seven characters long. The first character in the name must begin with an alphabetic character (A through Z, #, @ or $). The remaining characters can be any combination of characters (numeric, alphabetic and special). Commas(,), apostrophes('), quotation marks("), question mark(?), asterisks(*), and blanks are not allowed.

SET Specify the set identifier used to identify the entire set of files to be saved. This parameter is only valid if *ALL or a generic-name is specified on the FROMFILE parameter.

#SAVE: The files are saved as a Save All Set with a set identifier of #SAVE.

set-identifier: Specify the set identifier that is used to identify the entire set of file to be saved. Valid values range from 1 through 8 characters in length. The first character in the name must begin with an alphabetic character (A through Z, #, @ or $). The remaining characters can be any combination of characters (numeric, alphabetic and special). Commas(,), apostrophes('), quotation marks("), question marks(?), asterisks(*), and blanks are not allowed.

TOLABEL

Specifies the label given to the new tape or diskette file created by the save of a single file.

The TOLABEL name is also put into the System/36 file descriptor record which is stored at the beginning of the tape or diskette file. When the file is restored on a System/36, its name is the TOLABEL value (unless it is overridden by a //FILE OCL statement). Up to 8 characters can be used for the label.

If no value is specified, the value of the FROMFILE parameter is used as the diskette or tape label. If the FROMFILE parameter is an extended name, the characters between the quotation marks are used as the label. For example, specifying FROMFILE("MIKE&BOB") and no TOLABEL parameter would be the same as specifying TOLABEL('MIKE&BOB').

Note: This parameter cannot be specified if *ALL or generic*-name is specified on the FROMFILE parameter. The name of each file saved is used for the diskette or tape file label. If the name of the file is an extended name, the characters between the quotation marks are used as the label. For example, if the name of the file being saved is "A+B," the name of the diskette or tape file would be A+B.

CRTDATE

Specifies, for a date-differentiated file (maintained by the System/36 environment), which instance (member) of the file is saved. The default is to save the last instance (most recently created member) of the specified file. The date must be typed in the job date format (DATFMT). If separators, specified by the job date separator value (DATSEP), are used, the value must be enclosed in apostrophes. This parameter is valid only if a database physical file is being saved.

***LAST:** The most recently created member for the specified file is saved.

***ALL:** All members in the date-differentiated file are saved. If the file being saved is not date-differentiated, only the last member created in the file is saved. *ALL is valid only when saving multiple files and either *ALL or a generic file name is specified on the FROMFILE parameter.

file-creation-date: Specify the creation date of the date-differentiated file member to be saved. A file creation date can only be specified if a single file is to be saved and a file-name is specified on the FROMFILE parameter. (The date must be entered in the job date format (DATFMT); if separators, specified by the job date separator value (DATSEP), are used, the value must be enclosed in apostrophes.)

The date specified is converted to year/month/day format and the member of the file (specified by the FILE parameter) with the name ’Myymmdd’ is saved. For example, to save the instance of file ’FRED’ created on July 10, 1986, specify FROMFILE(FRED) CRTDATE(’07/10/86’), which would save member M860710 in file FRED.

This example allows a migrated System/36 user to identify the desired file/member very much like a System/36. However, it does not lend itself to letting the user save off a single member from an iSeries 400 system multi-member file. For example, the user who has a file named FRED with members BOB and JOE could not pick which member should be saved (the user would have to copy the member to a new file and save the new file specifying CRTDATE(*LAST) or rename the member to follow the ’Myymmdd’ name syntax).

SEQNBR

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

***END:** The system saves the specified file starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends. If the sequence number to be assigned to the specified file is greater than 9999, an error message is issued and the operation ends. If multiple files are to be saved, the next file is saved to a file after the first file saved, and so on. If the sequence number to be assigned ever exceeds 9999, an error message is issued and the operation ends.

file-sequence-number: Specify the sequence number of the file that is used. Valid values range from 1 through 9999.

If this sequence number already exists on the tape volume, the tape label at that sequence number must match the TOLABEL parameter. The existing file at that sequence number is overwritten, and all subsequent files on the volume are not accessible after the save.

If a new tape file is added to the tape, the sequence number must be one higher than the sequence number of the last tape file on that volume. No gaps are allowed in the series of sequence numbers.

If multiple files are being saved, this sequence number is used for the first file. All remaining files are saved as if *END was specified on the parameter SEQNBR. If the sequence number to be assigned ever exceeds 9999, an error message is issued and the operation ends.

VOL Specifies the volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in Commonly used parameters.

***MOUNTED:** The file to be saved is copied on whatever volume is on the device. For diskettes, use whatever volume is loaded in the diskette drive. If the diskette loaded in the diskette drive is full of active files, the save ends.

volume-identifier: Specify the volume identifiers of the tapes or diskettes used for saving the file. For each tape or diskette volume name, up to six characters can be specified using any combination of letters and numbers. The user should ensure that the volume name meets the System/36 volume identifier requirements. Up to 50 volume identifiers can be specified.

RETAIN

Specifies the retention period for the newly created tape or diskette file. The file is protected and cannot be written over until the day after the retention period ends.

1: The default retention period is one day.

retention-period: Specify the number of days the tape or diskette file should be kept. Valid values range from 0 through 999. If a retention period of 999 is specified, the tape or diskette file becomes a permanent file.

ENDOPT

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

Note: This parameter is ignored if a diskette device is specified in the DEV parameter.

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

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

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

PHYFILE

Specifies the qualified name of the file that receives the copied source file member data. This parameter is required if DEV(*PHYFILE) is specified.

If a file by this name does not exist, it is created as a non-keyed, program-described physical file with a record length of 256 bytes. If a file already exists by this name, it is used as long as it is a non-keyed physical file with a record length of 256 bytes.

The copied records are put in the first member of the physical file. If the file has no members, a member is created using the name syntax ’Myymmdd’ and the system date. For example, if the SAVS36LIBM was run on June 28, 1987, the member name would be M870628.

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

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

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

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

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

DTACPR

Specifies, when a diskette is used, whether the data is compressed into System/36 compatible format before it is written to the diskette. If the save command is operating while other jobs on the system are active and data compression is used, the overall system performance may be affected. This parameter is not valid if *PHYFILE or a tape device is specified on the DEV parameter.

***NO:** The data is not compressed before being written to the diskette.

***YES:** The data is compressed before being written to the diskette.

MBROPT

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

***NOREPLACE:** If a file already exists by the name specified on the PHYFILE parameter in the specified library, an error message is sent and the data in that member is not replaced.

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

Examples for SAVS36F

Example 1: Saving a Single File

```
SAVS36F  FROMFILE(PETE)  FROMLIB(QS36F)  DEV(I1)
```

This command saves the file named PETE located in library QS36F. Assuming that I1 is the name of a diskette device description, the file is saved on the diskette placed in the diskette drive. The diskette file label is PETE (same as the FROMFILE name). If PETE is a date-differentiated physical file, the most recently created instance (member) of PETE is saved. The diskette file has a retention period of one day (the retention period ends at midnight of the following day).

Example 2: Saving a Single File

```
SAVS36F  FROMFILE(MSTRPAY)  FROMLIB(PAYLIB)  DEV(T1 T2)
  TOLABEL('PAY.MSTR')  RETAIN(999)
```

The file named MSTRPAY located in library PAYLIB is saved. Assuming that T1 and T2 are tape devices, the file is copied to the tapes on devices T1 and T2. The tape file label is PAY.MSTR and the tape file is a permanent file. The last tape used for the save is rewound at the end of the save operation.

Example 3: Saving Multiple Files

```
SAVS36F  FROMFILE(*ALL)  FROMLIB(QS36F)  DEV(T1 T2)
  GROUP(*ALL) SET(ALLFL) RETAIN(999)
```

All database physical and logical files in library QS36F (including all files that belong to a file group) are saved. If any of the files are date-differentiated files, only the last member created in each file is saved. Assuming that T1 and T2 are tape devices, the files are copied to the tape volumes that are placed in tape drives T1 and T2. The label of the tape files created are the same as the names of the files that are saved. The first tape file created is located after the last sequence number on the tape. The remaining files are located after that first file. The tape files created are permanent. The last tape used for the save is rewound at the end of the save operation. The set identifier associated with this save all set is ALLFL.

Example 4: Saving Multiple Files

```
SAVS36F  FROMFILE(*ALL)  FROMLIB(QS36F)  DEV(T1 T2)
  GROUP(*NONE) CRTDATE(*LAST) SET(NOGFL) RETAIN(999)
```

All database physical and logical files in library QS36F except those files that belong to a file group are saved. If any of the files are date-differentiated files, only the last member created in each file is saved. Assuming that T1 and T2 are tape devices, the files are copied to the tape volumes that are placed in tape drives T1 and T2. The label of the tape files created is the same as the names of the files that are saved. The first tape file created is located after the last sequence number on the tape. The remaining files are located after that first file. The tape files created are permanent. The last tape used for the save is rewound at the end of the save operation. The set identifier associated with this save all set is NOGFL.

Example 5: Saving Multiple Files

```
SAVS36F  FROMFILE(*ALL)  FROMLIB(GRPLIB)  DEV(I1)
  GROUP(GRP) CRTDATE(*ALL)
```

All database physical and logical files in library GRPLIB that belong to file group GRP (GRP.01, GRP.02, and so on) are saved. If any of the files are date-differentiated files, all members in the files are saved. Assuming that I1 is a diskette drive, the files are copied to the diskette that is placed in the diskette drive. The label of the diskette files created is the same as the names of the files that are saved. The diskette files expire after one day. The set identifier associated with this save all set is #SAVE.

Example 6: Saving Multiple Files

```
SAVS36F  FROMFILE(PAY*)  FROMLIB(PAYROLL)  DEV(I1)
  SET(PAYSET) CRTDATE(*LAST) VOL(PAYDKT) RETAIN(10)
```

All database physical and logical files in library PAYROLL whose names begin with the characters PAY (PAY.01, PAYRATE, and so on) are saved. If any of the files are date-differentiated files, only the last member created is saved. Assuming that I1 is a diskette drive, the files are copied to a diskette with a volume identifier of PAYDKT. The label of the diskette files created is the same as the names of the files that are saved. The diskette files expire after ten days. The set identifier associated with this save all set is PAYSET.

Additional Considerations

A status message (CPI2C11) is sent when the processing begins for each database physical file that is saved. A completion message (CPC2C14 or CPC2C15) is sent after each file has been successfully saved.

Saving a Single File

A single file can be saved by specifying the name of the file on the FROMFILE parameter. If the file has multiple members (date-differentiated file), the user can:

- Specify which instance of the file (member) is to be saved by specifying the file’s creation date on the CRTDATE parameter.
- Specify that the most recently created file (last member) is to be saved by specifying *LAST on the CRTDATE parameter.

Note: *ALL cannot be specified on the CRTDATE parameter to save all instances of the file (all members) when saving a single file.

If the file is being saved to diskette or tape, the TOLABEL parameter can be used to specify the label of the new diskette or tape file created during the save operation. The label is also used when the file is restored.

If a single file is to be saved and a problem occurs that will not allow the operation to continue, an escape message is sent to the program message queue previous to the command processing program (CPP). This escape message informs the user of the cause and recovery of the problem. The message ends the operation.

Saving Multiple Files

Multiple files can be saved by specifying one of the following on the FROMFILE parameter:

- *ALL specifies the following:
  – If the value on the GROUP parameter is *ALL, all physical and logical files in the specified library (FROMLIB parameter) are saved, including all files that are part of file groups.
  – If the value on the GROUP parameter is *NONE, all physical and logical files in the specified library (FROMLIB parameter) are saved, except those files that are part of file groups.
  – If the value on the GROUP parameter is a file group name, all physical and logical files in the specified library (FROMLIB parameter) that are part of the specified file group are saved.
- generic*-file-name specifies that all physical and logical files in the specified library (FROMLIB parameter) with the same prefix as the generic name are saved.

When saving multiple files, a set identifier (SET parameter) is required. If a value is not specified, **#SAVE** is used.

The TOLABEL parameter cannot be specified. The label of the diskette or tape file created will be the same as the database file that is copied into it.

A specific file creation date cannot be specified on the CRTDATE parameter. If the file has multiple members (date-differentiated file), the most recently created member (CRTDATE(*LAST)) or all members (CRTDATE(*ALL)) can be saved.

If multiple files are to be saved and a problem is found that will not allow the operation to continue for a particular file, an inquiry message will be sent. If the SAVS36F job is interactive, the message is sent to the work station where the job was submitted. If the job is batch, the message is sent to the QSYSOPR message queue. The inquiry message informs the user of the cause and recovery of the problem. The messages have the following response options:

- C-Cancel the save operation.
- R-Retry saving the file after the problem has been corrected.
- I-Ignore saving this file and continue with the save of the next file.

The default response is I.

If multiple files are to be saved, and the operation was successful for all selected files, a completion message (CPC2C19) is sent after all files have been saved. This completion message informs the user the number of files that were saved. If any of the selected files cannot be saved, an escape message (CPF2C5B) is sent at the end of the save operation. The message informs the user of the number of files that were saved and the number that were not saved.

Error messages for SAVS36F

*ESCAPE Messages

CPF2C4A
Device &1 not correct for command.

CPF2C4B
Duplicate device &1 specified in device name list.

CPF2C4C
Diskette device &1 included in multiple device specification.

CPF2C4F
Diskette format not correct for DTACPR(*YES).

CPF2C47
Existing file &1 or member &3 in library &2 not replaced.

CPF2C48
Input file &1 in &2 not correct for command.

CPF2C49
Output file &1 in &2 not correct for command.

CPF2C5B
Not all files were saved.

CPF2C5C
Save operation ended before all files were saved.

CPF2C5D
No files saved from library &1.

CPF2C5E
Input file &1 in &2 not correct for command.

CPF2C5F
Tape file sequence numbers beyond 9999 not allowed.

CPF2C50
File description for file &1 is not available.

CPF2C51
Member information for file &1 in library &2 is not available.

CPF2C52
Error occurred during attempt to create file &1 in library &2.

CPF2C54
FROMFILE name &1 too long to use for TOLABEL parameter.

CPF2C55
TOLABEL parameter value &1 contains embedded blank(s).

CPF2C56
Physical file name &1 too long.

CPF2C58
Diskette format not acceptable for System/36.

CPF2C59
FROMFILE name &1 too long.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9814
Device &1 not found.

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

CPF9826
Cannot allocate file &2.

CPF9830
Cannot assign library &1.

CPF9831
Cannot assign device &1.

CPF9845
Error occurred while opening file &1.

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

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

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

SAVS36LIBM (Save System/36 Library Members) Command Description

SAVS36LIBM Command syntax diagram

Purpose

The Save System/36 Library Members (SAVS36LIBM) command creates a copy of source file members in a file on diskette or tape that can be restored on a System/36, or into a database physical file on the iSeries 400 that can be sent to a System/36.

The saved member file is formatted like a record-mode LIBRFILE created on a System/36 using the System/36 FROMLIBR system OCL procedure (or the equivalent OCL call of the $MAINT SSP utility). On a System/36, the diskette, tape, or database physical file can be restored using the TOLIBR system OCL procedure (or the equivalent OCL call of the $MAINT SSP utility).

If saving to a database physical file but the specified file (PHYFILE parameter) does not exist, it is created.

This command is intended only for exchanging source and procedure data with a System/36. It provides a simplified command interface for an iSeries 400 customer who migrated from a System/36, but is not well-suited for backing up of iSeries 400 source files. Use the iSeries 400 CL commands (SAVOBJ or SAVCHGOBJ) for creating a backup copy of an iSeries 400 source file or individual source file members.

Restrictions:

1. The following authorities are required when running on a system using resource security:
   - *USE authority for this command
   - *USE authority for the library specified in the FROMLIB parameter
   - *USE authority for file QS36SRC in the specified library if saving source library members
   - *USE authority for file QS36PRC in the specified library if saving procedure library members
   - *USE authority for the library specified in the PHYFILE parameter if saving to a physical file
   - *CHANGE authority for the library specified in the PHYFILE parameter if saving to a physical file and the file does not exist
   - *CHANGE and *OBJMGMT authority for that file if saving to a physical file with MBROPT(*ADD)
   - *ALL authority for the file if saving to a physical file with MBROPT(*REPLACE)
   - *USE authority for the diskette device description object, *USE authority for device file QSYSDKT in library QSYS if saving to diskette

- *USE authority for the tape device description object and *USE authority for device file QSYSTAP in library QSYS if saving to tape

2. All diskettes that are used to save the members should be initialized using the INZDKT CL command or the equivalent System/36 environment function (INIT OCL procedure or $INIT SSP utility). For a two-sided diskette, use a sector size of 256 or 1024. For a one-sided diskette, use a sector size of 128 or 512. If tape is used, each tape volume used should first be initialized with standard labels using the INZTAP CL command or the equivalent System/36 environment function (TAPEINIT OCL procedure or $TINIT SSP utility). Use a density of 1600 bits per inch when initializing the tape.

Note: If the tape or diskette has not been initialized as stated above, the System/36 will not be able to process the media.

   If a tape or diskette is used that has not been properly initialized, a message is sent to the system operator that allows the operator to cancel the save or initialize the volume and continue.

3. Object-level functions, other than read operations, should not be used for files QS36SRC and QS36PRC while members are being saved by SAVS36LIBM. If the necessary files cannot be allocated, no members are saved.

   Record-level functions, other than read operations, should not be used for members being saved. Concurrent activity against a member (for example, adding or removing records) can cause the member to be omitted from the save operation.

4. If saving a file to diskette, the diskettes used cannot contain an active file with the same name as the TOLABEL parameter value, because the iSeries 400 does not allow duplicate diskette file labels.

5. Only members from source files QS36SRC (for *SRC members) and QS36PRC (for *PRC members) in the specified library can be saved using the SAVS36LIBM command. Only the member data is saved from the source file member (that is, the source sequence number and change date fields are not saved).

6. The specified member name or generic member name (FROMMBR parameter) must meet iSeries 400 naming standards. When saving a member that has an extended name, the quotation mark characters are not stored as part of the member name in the output file. For example, member “A+B” would be saved as A+B.

Required Parameters

FROMMBR

Specifies the names of the members being saved. Specify a single member name or a single generic member name or *ALL for the members to be saved.

***ALL:** All members are saved of the member type specified on the SRCMBRS parameter.

member-name: Specify the names of the members being saved.

generic-member-name:* Specify the generic name of the member. 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.

FROMLIB

Specifies which library contains the members being saved.

DEV Specifies the names of the devices used for the save operation. Each device name must already be known on the system by a device description.

***PHYFILE:** The output file is the database physical file specified by the PHYFILE parameter.

device-name: Specify the name of the diskette device, the media library device, or the names of tape devices that are used for the operation. If more than one tape device is used, specify the names of the devices in the order in which they are used. When more than one tape volume is used, using more than one tape device permits one tape volume to be rewound or unloaded while another tape device processes the next tape volume. More information on device names is in the APPC, APPN, and HPR topic in the Information Center.

Optional Parameters

SRCMBRS

Specifies which member types (source and procedure members on System/36) are saved.

***ALL:** Save System/36 source and procedure members (from QS36SRC and QS36PRC) which match the member name specified (FROMMBR parameter). For example, if FROMMBR(PAY*) and SRCMBRS(*ALL) is specified, all members from both source files that begin with the letters 'PAY' are saved.

***SRC:** Only System/36 source members (from file QS36SRC), which match the member name specified (FROMMBR parameter), are saved.

***PRC:** Only System/36 OCL procedure members (from file QS36PRC), which match the member name specified (FROMMBR parameter), are saved.

RCDLEN

Specifies the file record length (in bytes) used when copying the members. This parameter is ignored when saving members to an existing database physical file.

120: This is the maximum record length allowed for System/36 source and procedure library members. Using this length ensures that no data is truncated when copying the records to the output file.

record-length: Specify the logical record length of the statements in the member. Valid values range from 40 through 120 and must be less than or equal to the record length of the file.

TOLABEL

Specifies the label of the diskette or tape file that receives the copied source file member data. Up to 8 characters can be used for the label value for the diskette or tape file. If the DEV parameter is not *PHYFILE, a TOLABEL value must be specified.

SEQNBR

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

***END:** The system saves the specified members starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends. If the sequence number to be assigned to the specified file is greater than 9999, an error message is issued and the operation ends.

file-sequence-number: Specify the sequence number of the file that is used. Valid values range from 1 through 9999.

If this sequence number already exists on the tape volume, the tape label at that sequence number must match the TOLABEL parameter value. The existing data file at that sequence number is overwritten, and all subsequent files on the volume are not accessible after the save operation.

If a new tape file is added to the tape, the sequence number must be one greater than the sequence number of the last tape file on that volume. No gaps are allowed in the series of sequence numbers.

VOL Specifies the volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in Commonly used parameters.

***MOUNTED:** The members saved are copied on the volume that is in the device. For diskettes, use the volume that is in the diskette drive. If the diskette loaded in the diskette drive is full of active files, the save operation ends.

volume-identifier: Specify the volume identifiers of the tapes or diskettes used for saving the members. For each tape or diskette volume name, up to 6 characters can be specified using any combination of letters and numbers. The user should ensure that the volume name meets the System/36 volume identifier requirements. Up to 50 volume identifiers can be specified.

RETAIN

Specifies the retention period for the newly created tape or diskette file. The file is protected and cannot be written over until the day after the retention period ends.

1: A retention period of one day is used.

retention-period: Specify the number of days the tape or diskette file should be kept. Valid values range from 0 through 999. If a retention period of 999 is specified, the tape or diskette file becomes a permanent file.

EXCHTYPE

Specifies the exchange type for the newly created diskette file.

***E:** The default is to create an E-exchange diskette file. An E-exchange file is a system save file that can be restored either on a System/36 (using the TOLIBR procedure) or on an iSeries 400 (using the RSTS36LIBM command).

***BASIC:** The output diskette file is to be in basic exchange format. A basic exchange format file can be restored or copied to a System/34 or System/32 or any other system that supports the basic exchange diskette format.

ENDOPT

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

Note: This parameter is ignored if a diskette device is specified in the DEV parameter.

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

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

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

PHYFILE

Specifies the qualified name of the file that receives the copied source file member data. This parameter is required if DEV(*PHYFILE) is specified.

If a file by this name does not exist, it is created in the current library if a library name was not specified, as a non-keyed, program-described physical file with the specified record length (RCDLEN parameter). If a file already exists by this name, it is used as long as it is a non-keyed physical file with a record length from 40 to 120.

The copied records are put in the first member of the physical file. If the file has no members, a member is created using the name syntax ’Myymmdd’ and the system date (for example, if the SAVS36LIBM was run on June 28, 1987, the member name would be M870628).

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

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

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

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

file-name: Specify the name of the file that receives the copied source file member data.

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.

Examples for SAVS36LIBM

Example 1: Saving Single Procedure Member

```
SAVS36LIBM  FROMMBR(XYZ1)  FROMLIB(JOHNSON)  DEV(I1)
  SRCMBRS(*PRC)  TOLABEL(XYZ1)
```

The single OCL procedure member XYZ1 (in source file QS36PRC) in library JOHNSON is saved. Assuming I1 is a diskette device, the member is saved to diskette file XYZ1. The file length is 120 and the retention period is one day.

Example 2: Saving All Source and Procedure Members

```
SAVS36LIBM  FROMMBR(X*)  FROMLIB(ORDER)  DEV(*PHYFILE)
  PHYFILE(NETLIB/S36SRC) MBROPT(*ADD)
```

All source and procedure members (members in QS36SRC and QS36PRC) with names starting with the character 'X' in library ORDER are saved into database physical file S36SRC in library NETLIB. The copied records are added after any records already in the file.

Error messages for SAVS36LIBM

*ESCAPE Messages

CPF2C4A

Device &1 not correct for command.

CPF2C4B

Duplicate device &1 specified in device name list.

CPF2C4C

Diskette device &1 included in multiple device specification.

CPF2C43

Saved &2 members from library &1, &3 members not saved.

CPF2C44

No members saved from library &1.

CPF2C48

Input file &1 in &2 not correct for command.

CPF2C49

Output file &1 in &2 not correct for command.

CPF2C5E
Input file &1 in &2 not correct for command.

CPF2C5F
Tape file sequence numbers beyond 9999 not allowed.

CPF2C50
File description for file &1 is not available.

CPF2C51
Member information for file &1 in library &2 is not available.

CPF2C52
Error occurred during attempt to create file &1 in library &2.

CPF2C55
TOLABEL parameter value &1 contains embedded blank(s).

CPF2C58
Diskette format not acceptable for System/36.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9814
Device &1 not found.

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

CPF9826
Cannot allocate file &2.

CPF9830
Cannot assign library &1.

CPF9845
Error occurred while opening file &1.

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

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

*STATUS Messages

CPI2C13
Copying records from file &1 in library &2 member &3.

SAVUSFCNR (Save Ultimedia System Facilities Container) Command Description

SAVUSFCNR Command syntax diagram

Purpose

The Save Ultimedia System Facilities Container (SAVUSFCNR) command can be used to save the following:

- Multimedia objects that have been copied to a shared folder container
- Document library objects (DLOs) that hold the Ultimedia Facilities data stream files.

You can transfer multimedia objects that reside in an Ultimedia Facilities container between systems by saving the multimedia objects on this system with this command and then restoring the multimedia objects on the target system with the Restore Ultimedia System Facilities Container (RSTUSFCNR) command.

Note: If you are saving the multimedia objects for archival or recovery purposes, use the Save Document Library Object (SAVDLO) command on the Ultimedia Facilities' base folder, QUMBREPO, instead of this command.

Restrictions: Only those objects that have been placed in an Ultimedia Facilities container or whose data stream files are in the Ultimedia Facilities repository are saved. If the data stream file is not in the Ultimedia Facilities repository, only the object's DLO is saved and a message is sent to the job log stating that the data stream file for the object was not saved.

Required Parameters

CNR Specifies the identifier of the existing Ultimedia Facilities container object that is to be saved. The identifier specified must be 20 characters.

DEV Specifies the name of the existing device on which the multimedia objects are to be saved.

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

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

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

Optional Parameters

VOL Specifies the volume identifiers of the volumes on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter is in commonly used parameters.

***MOUNTED:** The data is saved on the volumes placed in the device. For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are put on the device and used. Each volume identifier contains a maximum of 6 alphanumeric characters. A blank is used as a separator character when listing multiple identifiers. You can specify a maximum of 75 volumes.

SEQNBR

Specifies the tape sequence number used as the starting point for the save operation.

***END:** The system saves the multimedia object starting after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

file-sequence-number: Specify the sequence number of the file that is used as a starting point to save the multimedia objects.

LABEL

Specifies the label of the container object on the tape that is used for the save operation.

***CNR:** The container label is the same as the last 17 characters of the Ultimedia Facilities container identifier specified on the CNR parameter.

data-file-identifier: Specify the data file identifier that is used as the label of the data file for the save operation. A maximum of 17 characters can be specified.

EXPDATE

Specifies the expiration date. The files cannot be overwritten until the expiration date. The expiration date must be later than or equal to the current date.

TGTRLS

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the *target-release* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V3R6M0 is version 3, release 6, modification level 0.

Valid values depend on the current version, release, and modification level, and they change with each new release. See the **Valid Values for TGTRLS Parameter** table in the Backup, Recovery, and Availability topic in the Information Center for a complete list of valid values.

***CURRENT:** The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV:** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

target-release: Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

ENDOPT

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

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

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

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

SAVF Specifies the qualified name of the save file used to contain the save data. The save file must be empty or CLEAR(*ALL) must be specified.

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

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

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

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

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

CLEAR

Specifies whether uncleared tapes or save files encountered during the save operation are automatically cleared. An uncleared tape is one containing a file with an expiration date later than the date of the save operation.

Note: This parameter does not control initializing the tapes. The tapes used to perform the save operation should be initialized to a standard label format before the save command is issued. You can use the Initialize Tape (INZTAP) command and specify a value on the NEWVOL parameter to initialize a tape to a standard label format. If a tape volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the tape volume.

***NONE:** No save file or uncleared tape encountered during the save operation is automatically cleared. The save operation proceeds as follows:

- If a save file is not cleared, an inquiry message is sent to the work station message queue for an interactive job, or to the operator for a batch job.
- If the first tape is not cleared and enough space is available to start the operation, the save operation starts at the end of the existing data. If enough space is not available to start the save operation, a message is sent and the operation ends.
- If a tape after the first tape is not cleared and the operation cannot proceed, an inquiry message is sent. The operator can either end the save operation or specify that the currently selected tape be cleared so the operation can continue.

***ALL:** All uncleared tapes or any save file encountered during the save operation are cleared automatically.

If tapes are used and a sequence number is specified on the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER:** The uncleared tapes after the first tape are automatically cleared. If the operation cannot proceed because the first tape is not clear, an inquiry message is sent asking the system operator to end the operation or to specify that the currently selected tape be cleared so the operation can continue.

Note: This value is not valid for a save file.

SAVACT

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state, this parameter is ignored and the save operation is performed as if SAVACT(*NO) was specified.

_*NO:_ The objects that are in use by another job are not saved. Objects cannot be changed while they are being saved.

***YES:** The objects can be changed during the save operation.

SAVACTWAIT

Specifies the amount of time to wait for an object that is in use before continuing the save operation. If a lock is not obtained in the specified time, the object is not saved.

_120:_ The system waits a maximum of 120 seconds for an object lock before continuing the save operation.

***NOMAX:** The amount of time to wait has no maximum.

wait-time: Specify the time (in seconds) to wait for an object lock before continuing the save operation. Valid values range from 0 through 99999.

DTACPR

Specifies whether data compression is performed.

*DEV: If the tape device has the hardware compression feature installed, processing proceeds as if DTACPR(*YES) is specified. If the compression feature is not installed or if save data is written to a diskette or save file, processing proceeds as if DTACPR(*NO) is specified.

Note: If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***YES:** If the save operation is writing data to a tape device and the target device has the hardware compression feature, hardware compression is done. If the compression feature is not present, or if the save data is written to a save file, software data compression is done. If the save operation is being done while other jobs on the system are active and software data compression is used, the overall system performance may be affected.

***NO:** No data compression is done.

COMPACT

Specifies whether device data compaction is done.

*DEV: Device data compaction is done if the data is saved to tape and all tape devices specified on the DEV parameter support the compaction feature.

Note:

If *DEV is specified on both the DTACPR parameter and the COMPACT parameter, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified on the DTACPR parameter and *DEV is specified on the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***NO:** Device data compaction is not done.

Example for SAVUSFCNR

```
SAVUSFCNR   CNR(SEMINARVIDEOSPOT0004)  DEV(TAP01)
```

This command saves the shared folder container SEMINARVIDEOSPOT004 to the tape that is placed in the tape device named TAP01. Because this command uses the default values for the following attributes, the save operation begins after the last sequence number on the tape, the data file identifier is INARVIDEOSPOT0004, and the tape is rewound after the operation ends.

SAVRSTCHG (Save/Restore Changed Object) Command Description

SAVRSTCHG Command syntax diagram

Purpose

The Save/Restore Changed Object (SAVRSTCHG) command saves and concurrently restores a copy of each changed object, or group of objects located in the same library, to another system. For database files, only the changed members are saved and restored. Objects or members changed since the specified date and time are saved and restored.

Objects changed since the specified date and time are saved and restored with the following exceptions:

- If OBJJRN(*NO) is specified, database files currently being journaled are not saved and restored, unless journaling was started after the specified date and time. This ensures that changes made to a physical file before journaling starts are not lost (because they were not journaled in a journal receiver).
- Freed objects (programs, files, journal receivers, and so forth) are not saved.
- User-defined messages, job and output queue definitions, logical file definitions, and data queue descriptions are saved and restored, but the contents of those objects are not saved and restored. Logical file access paths are saved and restored if ACCPTH(*YES) is specified.

Specified objects that were changed, and the libraries where they reside, remain locked during the save and restore operation.

To determine the date and time that an object was changed, run the Display Object Description (DSPOBJD) command with DETAIL(*FULL) specified. For database file members that were changed, run the Display File Description (DSPFD) command.

The types of objects that can be saved and restored by this command are listed for the OBJTYPE parameter in Commonly used parameters. The system saves and concurrently restores the changed objects. The description of each object is changed with the date, time, and place when it was last saved. The UPDHST parameter controls this function.

Restrictions:

1. To use this command, the user must have the special authority *SAVSYS specified in the user profile by the SPCAUT parameter. Otherwise, the user must have object existence authority for each object specified, and execute authority to the specified library. If the user does not have the necessary authority to a specified object, all changed objects except that object are saved and restored.
2. No changed object that is being saved and restored can be changed by a job that is running when the save and restore operation occurs unless save-while-active is used.
3. When the contents of a save file are saved with SAVFDTA(*YES), the save file must be restored before objects contained in it can be restored.
4. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect/400 option is to be used, both systems must be joined by the OptiConnect/400 hardware and software.

Required Parameters

OBJ This parameter is the same as the OBJ parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ command to see a detailed parameter description.

LIB Specifies the name of the library that contains the changed objects to be saved and restored.

» ***ALLUSR**: All user libraries in the auxiliary storage pools (ASPs) defined by the ASPDEV parameter are searched. 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.

generic-library-name:* Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. Up to 300 generic library values can be specified. For more information on the use of generic functions, refer to generic functions.

library-name: Specify the name of the library to be saved and restored. Up to 300 library names can be specified.

RMTLOCNAME
Specifies the remote location to connect with. Specify the location name using the format cccccccc or nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name.

remote-location-name: Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name: Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Optional Parameters

OBJTYPE
This parameter is the same as the OBJTYPE parameter for the SAVCHGOBJ command. If you need further assistance, go to the SAVCHGOBJ to see a detailed parameter description.

OBJJRN
This parameter is the same as the OBJJRN parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

REFDATE
This parameter is the same as the REFDATE parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

REFTIME
This parameter is the same as the REFTIME parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

STRLIB
This parameter is the same as the STRLIB parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

OMITLIB
This parameter is the same as the OMITLIB parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

OMITOBJ
This parameter is the same as the OMITOBJ parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

» **ASPDEV**
This parameter is the same as the ASPDEV parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description. «

TGTRLS
This parameter is the same as the TGTRLS parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

PRECHK
This parameter is the same as the PRECHK parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

SAVACT

This parameter is the same as the SAVACT parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

SAVACTWAIT

This parameter is the same as the SAVACTWAIT parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

SAVACTMSGQ

This parameter is the same as the SAVACTMSGQ parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

ACCPTH

This parameter is the same as the ACCPTH parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

SAVFDTA

This parameter is the same as the SAVFDTA parameter for the SAVCHGOBJ command. If you need further assistance, go to SAVCHGOBJ to see a detailed parameter description.

OPTION

This parameter is the same as the OPTION parameter for the RSTOBJ command. If you need further assistance, go to the documentation for the RSTOBJ command in this book to see a detailed parameter description.

Note: The value *FREE is not valid for this command.

MBROPT

This parameter is the same as the MBROPT parameter for the RSTOBJ command. If you need further assistance, go to RSTOBJ to see a detailed parameter description.

ALWOBJDIF

This parameter is the same as the ALWOBJDIF parameter for the RSTOBJ command. If you need further assistance, go to RSTOBJ to see a detailed parameter description.

RSTLIB

Specifies the name of the library to which the objects are restored.

***LIB:** The restore library is the same name as the value specified on the LIB parameter.

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

» **RSTASPDEV**

This parameter is the same as the RSTASPDEV parameter for the RSTOBJ command. If you need further assistance, go to RSTOBJ to see a detailed parameter description. «

RSTASP

This parameter is the same as the RSTASP parameter for the RSTOBJ command. If you need further assistance, go to RSTOBJ to see a detailed parameter description.

FRCOBJCVN

This parameter is the same as the FRCOBJCVN parameter for the RSTOBJ command. If you need further assistance, go to RSTOBJ to see a detailed parameter description.

Examples for SAVRSTCHG

Example 1: Saving and Restoring Multiple Changed Objects

```
SAVRSTCHG   OBJ(*ALL)  RMTLOCNAME(SYSTEM1)
  LIB(*ALLUSR)  OBJTYPE(*PGM)
  STRLIB(MORE)  ALWOBJDIF(*NONE)
```

This command saves all changed objects whose type is *PGM and are located in user libraries beginning with the library named MORE. The objects are restored on the remote system named SYSTEM1.

Example 2: Saving and Restoring a Specific Changed Object

```
SAVRSTCHG   OBJ(PETE)  RMTLOCNAME(SYSTEM1)
  LIB(RON)  ALWOBJDIF(*ALL)  OPTION(*NEW)
```

This command saves changed objects named PETE of all types supported, which is located in the library named RON, and restores the object on the remote system named SYSTEM1 if the object does not already exist on the remote system.

Error messages for SAVRSTCHG

***ESCAPE Messages**

CPCAD80
&1 objects saved and restored.

CPFAD8D
An error occurred during the &1 operation.

CPFAD80
Unable to establish connection from &1 to &2.

CPFAD81
User profile &1 not found on remote location &2.

CPFAD82
Remote location &1 not found.

CPFAD83
Remote location &1 cannot be source location.

CPFAD84
ObjectConnect/400 internal error, function code &1, return code &2.

CPFAD86
Location name &1 unable to close &2.

CPFAD88
Unable to establish connection from &1 to &2.

CPFAD93
APPC failure. Failure code is &3.

CPF389C
ObjectConnect/400 internal error, function code &1, return code &2.

SAVRSTCFG (Save/Restore Configuration) Command Description

SAVRSTCFG Command syntax diagram

Purpose

The Save/Restore Configuration (SAVRSTCFG) command saves and restores configuration information without requiring a dedicated system. The information saved and restored includes the following:

- Configuration lists (*CFGL)
- Connection lists (*CNNL)
- Class-of-service descriptions (*COSD)
- Internet Packet Exchange descriptions (*IPXD)

- Mode descriptions (*MODD)
- NetBIOS descriptions (*NTBD)

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

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

Restrictions:

1. To use this command, the user must have *SAVSYS authority, or object existence authority for (or be the owner of) each object specified if the object already exists on the system.
2. This command is shipped with public *EXCLUDE authority.
3. The user must have either *ALLOBJ authority or authority to the command.
4. Changes made to the configuration while the SAVRSTCFG command is being run may not be reflected on the restore system, depending on when the changes occurred in relation to the send operation. If the operation cannot obtain a lock on a configuration object, a diagnostic message is issued; and the object is not saved and restored. The operation does not obtain the lock on a configuration object when a change or delete operation against the object does not complete before the default wait timer expires.
5. If any system resource manager objects are in use, a message is issued and none of them are saved and restored. The commands which can cause this to happen are Work with Hardware Products (WRKHDWPRD) and Work with Hardware Resources (WRKHDWRSC). No display commands can cause the system resource manager database to lock. All other known functions do read-only operations.
6. In order to save and restore token ring adapter's they must be done by themselves, with OBJ(*SRM) SRM(*TRA) and OBJTYPE(*ALL).
7. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect/400 option is to be used, both systems must be joined by the OptiConnect for OS/400 hardware and software.

Required Parameters

OBJ
Specifies the names of one or more objects to save and restore. This parameter is the same as the OBJ parameter for the RSTCFG command. If you need further assistance, go to the documentation for the RSTCFG command in this book to see a detailed parameter description.

RMTLOCNAME
Specifies the remote location to connect with. Specify the remote location name using the format cccccccc or nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name.

remote-location-name: Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name: Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Optional Parameters

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

***ALL:** All configuration object types that are specified by name are saved and restored. If OBJ(*ALL) is specified, all of the configuration objects are saved and restored.

***CFGL:** All configuration list object types are saved and restored.

***CNNL:** All connection list object types are saved and restored.

***COSD:** All class-of-service description objects types are saved and restored.

***IPXD:** All Internet Packet Exchange description object types are saved and restored.

***MODD:** All mode description object types are saved and restored.

***NTBD:** All NetBIOS description object types are saved and restored.

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

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

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

PRECHK

This parameter is the same as the PRECHK parameter for the SAVCFG command. If you need further assistance, go to the documentation for the SAVCFG command in this book to see a detailed parameter description.

ALWOBJDIF

This parameter is the same as the ALWOBJDIF parameter for the RSTCFG command. If you need further assistance, go to the documentation for the RSTCFG command in this book to see a detailed parameter description.

Examples for SAVRSTCFG

Example 1: Saving and Restoring Generic Objects

```
SAVRSTCFG   OBJ(ABCD*)  RMTLOCNAME(SYSTEM1)
  OBJTYPE(*CFGL)  ALWOBJDIF(*NONE)
```

This command saves and restores all objects whose names begin with ABCD and whose object type is configuration list (*CFGL). Only those objects that have no differences on SYSTEM1 will be restored.

Example 2: Saving and Restoring All Objects

```
SAVRSTCFG   OBJ(*ALL)  RMTLOCNAME(SYSTEM1)
```

This command saves and restores all configuration objects whose types match the five object types listed on the OBJTYPE parameter.

Error messages for SAVRSTCFG

*ESCAPE Messages

CPCAD80

&1 objects saved and restored.

CPFAD8D

An error occurred during the &1 operation.

CPFAD80

Unable to establish connection from &1 to &2.

CPFAD81

User profile &1 not found on remote location &2.

CPFAD82

Remote location &1 not found.

CPFAD83
Remote location &1 cannot be source location.

CPFAD84
ObjectConnect/400 internal error, function code &1, return code &2.

CPFAD86
Location name &1 unable to close &2.

CPFAD88
Unable to establish connection from &1 to &2.

CPFAD93
APPC failure. Failure code is &3.

CPF389C
ObjectConnect/400 internal error, function code &1, return code &2.

SAVRSTDLO (Save/Restore Document Library Object) Command Description

SAVRSTDLO Command syntax diagram

Purpose

The Save/Restore Document Library Object (SAVRSTDLO) command saves and restores the following objects to another system. The system must have a supported communication link with the system that the objects are being restored to.

- Documents
- Folders
- Distribution objects (mail)

Notes:

1. When a folder is saved and restored, the folder object is transferred along with the documents contained in that folder and the subfolders and documents in the subfolders and all successively nested folders and documents. Specific folders can be saved and restored individually using the DLO(*FLRLVL) parameter.
2. Distribution objects (mail) cannot be sent for individual users. Mail can be saved and restored for all users only.
3. SAVRSTDLO does not require a dedicated system; however, individual objects in use when the save and restore command is issued cannot be saved and restored. To ensure all document library objects are saved and restored, run this command when no office activity is occurring on the system.

Restrictions:

1. You must have *ALLOBJ or *SAVSYS special authority to use the following parameter combinations on this command:
   - DLO(*ALL) FLR(*ANY)
   - DLO(*CHG)
   - DLO(*MAIL)
   - DLO(*SEARCH) OWNER(*ALL)
   - DLO(*SEARCH) OWNER(*user-profile-name*)

   where the *user profile name* specified is not the *user profile name* of the user issuing the SAVRSTDLO command.
2. Users that do not have *ALLOBJ or *SAVSYS special authority must:

- Have *ALL authority for each document or folder to be sent.
- Be enrolled as Document Interchange Architecture (DIA) users.

3. Determining document or folder ownership does not include checking group profiles if one is associated with the specified user profile.
4. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect/400 option is to be used, both systems must be joined by the OptiConnect/400 hardware and software.

Required Parameters

DLO Specifies the document library objects to save and restore.

Note: System object name is not supported.

This parameter is the same as the DLO parameter for the SAVDLO command. If you need further assistance, go to the documentation for the SAVDLO command in this book to see a detailed parameter description.

RMTLOCNAME
Specifies the remote location to connect with. Specify the remote location name using the format cccccccc or nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name.

remote-location-name: Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name: Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Optional Parameters

FLR Specifies the name of the folder containing the document library objects to be saved and restored.

Note: System object name is not supported.

This parameter is the same as the FLR parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

SRCHTYPE
This parameter is the same as the SRCHTYPE parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

CHKEXP
This parameter is the same as the CHKEXP parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

CRTDATE
This parameter is the same as the CRTDATE parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

DOCCLS
This parameter is the same as the DOCCLS parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

OWNER
This parameter is the same as the OWNER parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

REFCHGDATE
This parameter is the same as the REFCHGDATE parameter for the SAVDLO command. If you need further assistance, go to SAVDLO to see a detailed parameter description.

REFCHGTIME
This parameter is the same as the REFCHGTIME parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

DOCL This parameter is the same as the DOCL parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

OMITFLR
This parameter is the same as the OMITFLR parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

TGTRLS
This parameter is the same as the TGTRLS parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

STG This parameter is the same as the STG parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

CMDCHRID
This parameter is the same as the CMDCHRID parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

SAVACT
This parameter is the same as the SAVACT parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

SAVACTWAIT
This parameter is the same as the SAVACTWAIT parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

ASP This parameter is the same as the ASP parameter for the SAVDLO command. If you need further assistance, go to SAVDLO command to see a detailed parameter description.

NEWOBJ
This parameter is the same as the NEWOBJ parameter for the RSTDLO command. If you need further assistance, go to RSTDLO to see a detailed parameter description.

ALWOBJDIF
This parameter is the same as the ALWOBJDIF parameter for the RSTDLO command. If you need further assistance, go to RSTDLO command to see a detailed parameter description.

RSTASP
This parameter is the same as the RSTASP parameter for the RSTDLO command.

Note: The default value of *SAVASP will cause the value specified for the ASP parameter to be used.

If you need further assistance, go to RSTDLO command to see a detailed parameter description.

Examples for SAVRSTDLO

Example 1: Saving and Restoring All Document Library Objects

```
SAVRSTDLO   DLO(*ALL)  RMTLOCNAME(SYSTEM1)
  FLR(*ANY)  SAVACT(*YES)  NEWOBJ(*NEW)
```

This command saves all document library objects located in any folder and restores the objects to a remote system named SYSTEM1. The objects can be changed during the save and restore operation and new names will be given to the libraries in which the objects are restored.

Example 2: Saving and Restoring Specific Document Library Objects

```
SAVRSTDLO   DLO(*SEARCH)  RMTLOCNAME(SYSTEM2)
  FLR(FLR2)  OWNER(USER8)  SRCHTYPE(*DOC)
```

This command searches for a folder named FLR2 and checks to see if the documents owned by user USER8. The folder and the documents in the folder are saved and restored on a remote system named SYSTEM2.

Error messages for SAVRSTDLO

*ESCAPE Messages

CPCAD82
&1 document library objects saved and restored.

CPFAD8C
An error occurred during the SAVRSTDLO operation.

CPFAD80
Unable to establish connection from &1 to &2.

CPFAD81
User profile &1 not found on remote location &2.

CPFAD82
Remote location &1 not found.

CPFAD83
Remote location &1 cannot be source location.

CPFAD84
ObjectConnect/400 internal error, function code &1, return code &2.

CPFAD86
Location name &1 unable to close &2.

CPFAD88
Unable to establish connection from &1 to &2.

CPFAD93
APPC failure. Failure code is &3.

CPF389C
ObjectConnect/400 internal error, function code &1, return code &2.

SAVRSTLIB (Save/Restore Library) Command Description

SAVRSTLIB Command syntax diagram

Purpose

The Save/Restore Library (SAVRSTLIB) command allows the user to save and restore a copy of one or more libraries to another system. The system must have a supported communication link with the restoring system.

Documents and folders contained in the QDOC library can be saved and restored by using the Save/Restore Document Library Object (SAVRSTDLO) command.

The SAVRSTLIB command saves and restores the entire library; this includes the library description, the object descriptions, and the contents of the objects in the library. For job queues, message queues, output queues, data queues, and logical files, only the object definitions are saved and restored, not the contents. Logical file access paths can be saved and restored by specifying ACCPTH(*YES). The contents of a save file can be saved by using the Save Save File Data (SAVSAVFDTA) command. By specifying SAVFDTA(*YES) on the SAVRSTLIB command, the contents of a save file can be saved and restored.

The libraries and their objects are not affected on the system unless the command specifies that the storage is to be freed. If a group of libraries is saved by specifying *NONSYS, *ALLUSR, or *IBM for the LIB parameter, the date, time, and place are updated in the history information for a data area in QSYS (data area QSAVLIBALL, QSAVALLUSR, or QSAVIBM).

The types of objects saved and restored by this command are the same as those listed for the OBJTYPE parameter in with the addition of *DTADCT. Certain OS/400 system objects that are not contained in user libraries (such as user profiles) are not saved and restored by this command. They can be saved by the Save System (SAVSYS) or Save Security Data (SAVSECDTA) commands, and restored by using the Restore User Profile (RSTUSRPRF) command.

Restrictions:

1. To use this command, the user must have either the special authority *SAVSYS specified in the user profile by the SPCAUT parameter, or the user must have:
   - Read authority for, or be the owner of, each library specified.
   - Object existence authority for each object in the library.

   If the user does not have the correct authorities for all of the libraries and objects specified, only those for which the user does have authority are saved and restored.
2. No library that is being saved and restored, or the objects in the library being saved and restored, can be updated by a job that is running at the time the save and restore operation occurs unless save-while-active (SAVACT) is used.
3. When the contents of a save file are saved and restored by specifying SAVFDTA(*YES), the save file must be restored before objects contained in it can be restored.
4. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect/400 option is to be used, both systems must be joined by the OptiConnect for OS/400 hardware and software.

Required Parameters

LIB Specifies which libraries to save and restore.

Note: If using *NONSYS, or *IBM keywords some libraries will not be saved and restored, because they are being used for this operation.

Note: If the user specifies *ALLUSR on this parameter, this command should be run when the specified libraries are not being used. If objects in a library are in use while the library is being saved and restored, the objects are not saved and restored. Some subsystems cannot be ended when using the SAVRSTLIB command. The libraries associated with these subsystems will not be saved and restored using the SAVRSTLIB command.

```
QSOC - if using the optical bus transport
       with the SAVRSTLIB command
QCMN - if using the communications transport
       with the SAVRSTLIB command
```

***NONSYS:** All user-created libraries, the QGPL and QUSRSYS libraries, and licensed program libraries such as QRPG and QIDU are saved. When *NONSYS is specified, the libraries are saved in alphabetical order on the media.

» ***ALLUSR**: All user libraries in the auxiliary storage pools (ASPs) defined by the ASPDEV parameter are searched. 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.

***IBM:** Saves all system (IBM) libraries except for the following Q libraries: »

```
QDOC        QRCLxxxxx   QSYS2       QUSRINFSKR
QDOCxxxx    QRCYxxxxx   QSYS2xxxxx  QUSRNOTES
QDSNX       QRECOVERY   QS36F       QUSROND
QGPL        QRPLOBJ     QTEMP       QUSRPOSGS
QGPL38      QRPLxxxxx   QUSER38     QUSRPOSSA
QMPGDATA    QSPL        QUSRADSM    QUSRPYMSVR
```

QMQMDATA
QMQMPROC
QPFRDATA
QRCL
QSPLxxxx
QSRV
QSYS
QSYSxxxxx
QUSRBRM
QUSRDIRCL
QUSRDIRDB
QUSRIJS
QUSRRDARS
QUSRSYS
QUSRVI
QUSRVxRxMx

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

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

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

library-name: Specify the names of the library to be saved and restored. A maximum of 300 library names can be specified.

generic-library-name:* Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. A maximum of 300 generic library names can be specified. For more information on the use of generic functions, refer to generic functions.

RMTLOCNAME

Specifies the remote location to connect with. Specify the remote location name using the format cccccccc or nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name.

remote-location-name: Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name: Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Optional Parameters

STRLIB

This parameter is the same as the STRLIB parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description.

OMITLIB

This parameter is the same as the OMITLIB parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description.

OMITOBJ

This parameter is the same as the OMITOBJ parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description.

ASPDEV

This parameter is the same as the ASPDEV parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description.

TGTRLS
This parameter is the same as the TGTRLS parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description.

PRECHK
This parameter is the same as the PRECHK parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description.

SAVACT
This parameter is the same as the SAVACT parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description.

SAVACTWAIT
This parameter is the same as the SAVACTWAIT parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description.

SAVACTMSGQ
This parameter is the same as the SAVACTMSGQ parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description.

ACCPTH
This parameter is the same as the ACCPTH parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description.

SAVFDTA
This parameter is the same as the SAVFDTA parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description. and restored.

STG This parameter is the same as the STG parameter for the SAVLIB command. If you need further assistance, go to SAVLIB to see a detailed parameter description.

OPTION
This parameter is the same as the OPTION parameter for the RSTLIB command. If you need further assistance, go to RSTLIB to see a detailed parameter description.

MBROPT
This parameter is the same as the MBROPT parameter for the RSTLIB command. If you need further assistance, go to RSTLIB to see a detailed parameter description.

ALWOBJDIF
This parameter is the same as the ALWOBJDIF parameter for the RSTLIB command. If you need further assistance, go to RSTLIB to see a detailed parameter description.

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

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

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

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

RSTASPDEV
This parameter is the same as the RSTASPDEV parameter for the RSTLIB command. If you need further assistance, go to RSTLIB to see a detailed parameter description.

RSTASP
This parameter is the same as the RSTASP parameter for the RSTLIB command. If you need further assistance, go to RSTLIB to see a detailed parameter description.

FRCOBJCVN
This parameter is the same as the FRCOBJCVN parameter for the RSTLIB command. If you need further assistance, go to RSTLIB to see a detailed parameter description.

Examples for SAVRSTLIB

Example 1: Saving and Restoring All User Libraries

```
SAVRSTLIB   LIB(*ALLUSR)  RMTLOCNAME(SYSTEM1)
  STRLIB(GFM1)  OPTION(*OLD)
```

This command saves all user libraries beginning with the GFM1 library and restores them on a remote system named SYSTEM1. Only the objects in the library having a saved version are restored.

Example 2: Saving and Restoring Specific Libraries

```
SAVRSTLIB   LIB(GRUNBOK TIMON VASEK)  RMTLOCNAME(SYSTEM1)
```

This command saves the following libraries GRUNBOK, TIMON and VASEK and restores them on a remote system named SYSTEM1.

Error messages for SAVRSTLIB

*ESCAPE Messages

CPCAD81
&1 libraries saved and restored.

CPFAD8B
An error occurred during the SAVRSTLIB operation.

CPFAD80
Unable to establish connection from &1 to &2.

CPFAD81
User profile &1 not found on remote location &2.

CPFAD82
Remote location &1 not found.

CPFAD83
Remote location &1 cannot be source location.

CPFAD84
ObjectConnect/400 internal error, function code &1, return code &2.

CPFAD86
Location name &1 unable to close &2.

CPFAD88
Unable to establish connection from &1 to &2.

CPFAD93
APPC failure. Failure code is &3.

CPF389C
ObjectConnect/400 internal error, function code &1, return code &2.

SAVRSTOBJ (Save/Restore Object) Command Description

SAVRSTOBJ Command syntax diagram

Purpose

The Save/Restore Object (SAVRSTOBJ) command saves and restores a single object, or a group of objects located in the same library, to another system. The objects can be restored to the same library or a different library. The types of objects that can be saved and restored are the same set of objects allowed on the Save Object (SAVOBJ) command. Objects on the system from which the objects are being saved are not affected by the SAVRSTOBJ command.

For job queues, output queues, data queues, message queues, and logical files, only the object descriptions are restored, not the contents. However, logical file access paths can be restored by specifying ACCPTH(*YES).

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

Restrictions:

1. To use this command, you must have either the special authority *SAVSYS specified in the user profile by the SPCAUT parameter or have (a) object existence authority for each object specified and (b) read authority for the specified library. If you do not have the necessary authority to a specified object, all objects except that one, are saved and restored.
2. Both systems intended to participate in the save and restore operation must be connected to the same APPN network, or if the OptiConnect/400 option is to be used, both systems must be joined by the OptiConnect for OS/400 hardware and software.

Required Parameters

OBJ This parameter is the same as the OBJ parameter for the SAVOBJ command. If you need further assistance, go to the documentation for the SAVOBJ command in this book to see a detailed parameter description.

LIB Specifies the library containing the objects to be saved from. Only one library name can be specified.

» ***ALLUSR**: All user libraries in the auxiliary storage pools (ASPs) defined by the ASPDEV parameter are searched. 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
» QSYS2xxxxx«
QS36F
QUSER38
QUSRADSM
QUSRBRM
QUSRDIRCL
QUSROND
QUSRPOSGS
QUSRPOSSA
QUSRPYMSVR
QUSRRDARS
QUSRSYS

QPFRDATA
QRCL
» QRCLxxxxx«
» QSYS2«
QUSRDIRDB
QUSRIJS
QUSRINFSKR
QUSRNOTES
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.

generic-library-name:* Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. Up to 300 generic library values can be specified. For more information on the use of generic functions, refer to generic functions.

library-name: Specify the name of the library to be saved and restored. Up to 300 library names can be specified.

RMTLOCNAME
Specifies the remote location to connect with. Specify the remote location name where cccccccc is the remote location name.

remote-location-name: Specify the remote location name associated with the system to which you want to restore objects.

Optional Parameters

OBJTYPE
This parameter is the same as the OBJTYPE parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

STRLIB
This parameter is the same as the STRLIB parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

OMITLIB
This parameter is the same as the OMITLIB parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

» **ASPDEV**
This parameter is the same as the ASPDEV parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description. «

TGTRLS
This parameter is the same as the TGTRLS parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

PRECHK
This parameter is the same as the PRECHK parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

SAVACT
This parameter is the same as the SAVACT parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

SAVACTWAIT

This parameter is the same as the SAVACTWAIT parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

SAVACTMSGQ

This parameter is the same as the SAVACTMSGQ parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

FILEMBR

This parameter is the same as the FILEMBR parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

ACCPTH

This parameter is the same as the ACCPTH parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

SAVFDTA

This parameter is the same as the SAVFDTA parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

STG

This parameter is the same as the STG parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

OMITOBJ

This parameter is the same as the OMITOBJ parameter for the SAVOBJ command. If you need further assistance, go to SAVOBJ to see a detailed parameter description.

OPTION

This parameter is the same as the OPTION parameter for the RSTOBJ command. If you need further assistance, go to RSTOBJ to see a detailed parameter description.

Note: The value *FREE is not valid for this command.

MBROPT

This parameter is the same as the MBROPT parameter for the RSTOBJ command. If you need further assistance, go to RSTOBJ to see a detailed parameter description.

ALWOBJDIF

This parameter is the same as the ALWOBJDIF parameter for the RSTOBJ command. If you need further assistance, go to RSTOBJ to see a detailed parameter description.

RSTLIB

Specifies the name of the library in which the objects are to be restored.

***LIB:** The name of the library that was specified on the LIB parameter is used.

restore-library-name: Specify the name of the library in which you want to restore the objects.

RSTASPDEV

This parameter is the same as the RSTASPDEV parameter for the RSTOBJ command. If you need further assistance, go to RSTOBJ to see a detailed parameter description.

RSTASP

This parameter is the same as the RSTASP parameter for the RSTOBJ command. If you need further assistance, go to RSTOBJ to see a detailed parameter description.

FRCOBJCVN

This parameter is the same as the FRCOBJCVN parameter for the RSTOBJ command. If you need further assistance, go to RSTOBJ to see a detailed parameter description.

Examples for SAVRSTOBJ

Example 1: Saving and Restoring Generic Objects

```
SAVRSTOBJ   OBJ(ABCD*)  LIB(ACE)
  RMTLOCNAME(SYSTEM1)
  OBJTYPE(*PGM)  ALWOBJDIF(*NONE)
```

This command saves the objects beginning with the characters ABCD located in the library named ACE and restores them on the remote system named SYSTEM1.

Example 2: Saving and Restoring a Specific Object

```
SAVRSTOBJ   OBJ(FLETCH)  LIB(CHASE)
  RMTLOCNAME(SYSTEM1)
  ALWOBJDIF(*ALL)  OPTION(*NEW)
```

This command saves the object named FLETCH located in the library named CHASE and restores it on the remote system named SYSTEM1 if it is does not already exist on the remote system.

Error messages for SAVRSTOBJ

***ESCAPE Messages**

CPCAD80
&1 objects saved and restored.

CPFAD8D
An error occurred during the &1 operation.

CPFAD80
Unable to establish connection from &1 to &2.

CPFAD81
User profile &1 not found on remote location &2.

CPFAD82
Remote location &1 not found.

CPFAD83
Remote location &1 cannot be source location.

CPFAD84
ObjectConnect/400 internal error, function code &1, return code &2.

CPFAD86
Location name &1 unable to close &2.

CPFAD88
Unable to establish connection from &1 to &2.

CPFAD93
APPC failure. Failure code is &3.

CPF389C
ObjectConnect/400 internal error, function code &1, return code &2.

SAVRST (Save/Restore Objects) Command Description

SAVRST Command syntax diagram

Purpose

The Save/Restore Objects (SAVRST) command saves and restores a copy of one or more objects that can be used in the integrated file system.

For more information about integrated file system commands, see the Integrated File System Introduction topic in the Information Center.

The SAVRST command uses the current save and restore support for objects in libraries and for document library objects. As a result, there are restrictions when you use the SAVRST command on these objects.

Restrictions:

1. For name patterns in the root directory:
    a. OBJ must be one of the following:
        - OBJ(('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT))
        - OBJ(('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT) ('/other values' *OMIT)
2. For names involving objects in libraries:
    a. OBJ must have only one name, except when omitting /QSYS.LIB from /*
    b. OBJ must be one of the following:
        - OBJ('/QSYS.LIB/libname.LIB')
        - OBJ('/QSYS.LIB/libname.LIB/*')
        - OBJ('/QSYS.LIB/libname.LIB/*.type')
        - OBJ('/QSYS.LIB/libname.LIB/objname.type')
        - OBJ('/QSYS.LIB/libname.LIB/filename.FILE/*')
        - OBJ('/QSYS.LIB/libname.LIB/filename. FILE/*.MBR')
        - OBJ('/QSYS.LIB/libname.LIB/filename. FILE/membername.MBR')
        - OBJ('/QSYS.LIB/*.type')
        - OBJ('/QSYS.LIB/objname.type')
        - OBJ('/QSYS.LIB/filename.FILE/*')
        - OBJ('/QSYS.LIB/filename.FILE/*.MBR')
        - OBJ('/QSYS.LIB/filename. FILE/membername.MBR')

    c. The .type must be an object type supported by SAVOBJ and RSTOBJ
    d. libname cannot be » QDOC, QDOCxxxx, QRCYxxxxx, QRECOVERY, QRPLOBJ, QRPLxxxxx, QSPL, QSPLxxxx, QSRV, QSYS, QSYSxxxxx, QTEMP, or QSR if libname.LIB is the last component of the name «
    e. SUBTREE must be *ALL
    f. For SAVRST:
        - CHGPERIOD end date and end time must be *ALL
        - CHGPERIOD must be default if a file member is specified
        - An object cannot be renamed
        - For database file members, OPTION(*NEW) only restores members for new files
3. For names involving document library objects:
    a. OBJ must have only one name, except when omitting /QDLS from /*
    b. OBJ and SUBTREE must be one of the following:
        - OBJ('/QDLS/path/foldername') SUBTREE(*ALL)
        - OBJ('/QDLS/path/documentname') SUBTREE(*OBJ)

    c. For SAVRST:
        - The defaults must be taken on the PRECHK and SAVACTMSGQ parameters
        - CHGPERIOD must be default with OBJ('/QDLS/path/documentname') SUBTREE(*OBJ)
        - CHGPERIOD start date cannot be *LASTSAVE

- CHGPERIOD end date and end time must be *ALL
- SAVACT cannot be *SYNC
- SAVACTMSGQ must be *NONE
- ALWOBJDIF must be *NONE or *ALL
- OPTION must be *ALL

4. Both systems intended to participate in the save and restore operation must be connected to the same APPN network, or, if the OptiConnect/400 option is to be used, both systems must be joined by the OptiConnect for OS/400 hardware and software.

Required Parameter

RMTLOCNAME

Specifies the remote location to connect with. Specify the remote location name using the format cccccccc or nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name.

remote-location-name: Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name: Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Optional Parameters

OBJ Specify the objects to save and restore.

Note: The objects will be restored with the same name.

This parameter is the same as the OBJ parameter for the SAV command. If you need further assistance, go to the documentation for the SAV command in this book to see a detailed parameter description.

SUBTREE

This parameter is the same as the SUBTREE parameter for the SAV command. If you need further assistance, go to the documentation for the SAV command in this book to see a detailed parameter description.

CHGPERIOD

This parameter is the same as the CHGPERIOD parameter for the SAV command. If you need further assistance, go to the documentation for the SAV command in this book to see a detailed parameter description.

TGTRLS

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the *target-release* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V3R6M0 is version 3, release 6, modification level 0.

Valid values depend on the current version, release, and modification level, and they change with each new release. See the **Valid Values for TGTRLS Parameter** table in the Backup, Recovery, and Availability topic in the Information Center for a complete list of valid values.

***CURRENT:** The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV:** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

target-release: Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

PRECHK

This parameter is the same as the PRECHK parameter for the SAV command. If you need further assistance, go to the documentation for the SAV command in this book to see a detailed parameter description.

SAVACT

This parameter is the same as the SAVACT parameter for the SAV command. If you need further assistance, go to the documentation for the SAV command in this book to see a detailed parameter description.

SAVACTMSGQ

This parameter is the same as the SAVACTMSGQ parameter for the SAV command. If you need further assistance, go to the documentation for the SAV command in this book to see a detailed parameter description.

» **ASPDEV**

This parameter is the same as the ASPDEV parameter for the SAV command. If you need further assistance, go to the documentation for the SAV command in this book to see a detailed parameter description. «

OPTION

This parameter is the same as the OPTION parameter for the RST command. If you need further assistance, go to the documentation for the RST command in this book to see a detailed parameter description.

ALWOBJDIF

This parameter is the same as the ALWOBJDIF parameter for the RST command. If you need further assistance, go to the documentation for the RST command in this book to see a detailed parameter description.

FRCOBJCVN

This parameter is the same as the FRCOBJCVN parameter for the RST command. If you need further assistance, go to the documentation for the RST command in this book to see a detailed parameter description.

Examples for SAVRST

Example 1: Saving and Restoring a Member Object

```
SAVRST   RMTLOCNAME(SYSTEM1)
  OBJ(('QSYS.LIB/JTEMP.LIB/ZXC.FILE/QYYCPDGT.MBR'))
```

This command saves the QYYCPDGT member object from the ZXC file, in library JTEMP and restores the object on the remote location named SYSTEM1.

Example 2: Saving and Restoring a Directory

```
SAVRST   RMTLOCNAME(SYSTEM2)
  OBJ(('MYDIR'))
  SAVACT(*YES)
  SAVACTMSGQ('QSYS.LIB/SVRTEST.LIB/ZXC.MSGQ')
```

This command saves the MYDIR directory while active, and will use the ZXC message queue to save messages.

Error messages for SAVRST

***ESCAPE Messages**

CPCAD80
&1 objects saved and restored.

CPFAD8D
An error occurred during the &1 operation.

CPFAD80
Unable to establish connection from &1 to &2.

CPFAD81
User profile &1 not found on remote location &2.

CPFAD82
Remote location &1 not found.

CPFAD83
Remote location &1 cannot be source location.

CPFAD84
ObjectConnect/400 internal error, function code &1, return code &2.

CPFAD86
Location name &1 unable to close &2.

CPFAD88
Unable to establish connection from &1 to &2.

CPFAD93
APPC failure. Failure code is &3.

CPF389C
ObjectConnect/400 internal error, function code &1, return code &2.

SLTCMD (Select Command) Command Description

SLTCMD Command syntax diagram

Purpose

The Select Command (SLTCMD) command produces a list of commands from a user-specified subset of command names. From this list, only the option to select the command is available.

Restrictions:

1. Only the libraries to which the user has USE authority will be searched.
2. Only the commands to which the user has some authority will be shown on the display.
3. To perform operations on the commands, the user must have USE authority to the command used by the operation, and the appropriate authority to the commands on which the operation is to be performed.

Required Parameter

CMD Specifies the qualified name of the commands being shown on the Select Command display.

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

***LIBL:** All libraries in the job's library list are searched. All objects in these libraries with the specified object name are shown.

***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.

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

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

***ALL:** All commands in the libraries specified are listed on the Select Command display.

generic-command-name:* Specify the generic name of the command being listed. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be selected only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

command-name: Specify the name of the command being listed.

Example for SLTCMD

```
SLTCMD  CMD(QSYS/*ALL)
```

This command shows a list of all commands in library QSYS. The option to prompt and run commands is available.

SNDBRKMSG (Send Break Message) Command Description

SNDBRKMSG Command syntax diagram

Purpose

The Send Break Message (SNDBRKMSG) command is used by the system operator to send an immediate message to one or more work station message queues. An immediate message is a message that is not predefined and is not stored in a message file. The command causes the message to be delivered always in break mode. The DSPMSG display is shown for the message when it is received, regardless of the setting of the message queue’s delivery mode, severity, and break handling program. However, the message may not be displayed in some cases, depending on the BRKMSG job attribute. This command is primarily intended for the system operator’s use.

Restrictions:

1. This command can be used to send break messages to work station message queues only.
2. This command cannot send inquiry messages (specified by MSGTYPE(INQ)) to multiple work stations.

Required Parameters

MSG Specifies the message text that is sent. A maximum of 512 characters can be specified. The message must be enclosed in apostrophes if it contains blanks or other special characters.

Coded Character Set Identifier (CCSID) Considerations

The text supplied on the MSG parameter is assumed to be in the CCSID of the job running this command unless a coded character set identifier is supplied on the CCSID parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center.

TOMSGQ

Specifies the qualified names of one or more work station message queues to which the break message is sent. Only the names of work station message queues can be specified.

***ALLWS:** The break message is sent to all work stations and Personal Computer message queues. This value cannot be specified if *INQ is specified on the MSGTYPE parameter.

The name of the message 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.

message-queue-name: Specify the name of the message queue to which the break message is being sent.

Optional Parameters

MSGTYPE

Specifies the type of message that is sent in break mode. Only informational or inquiry message types can be specified.

***INFO:** An information-only message is sent in break mode.

***INQ:** An inquiry message is sent in break mode; the work station receiving the message is expected to reply to it. Inquiry messages can be sent to only one work station at a time.

RPYMSGQ

Specifies, only if an inquiry message is sent, the name of the message queue to which the work station user's reply is sent.

The name of the message 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.

QSYSOPR: The replies to the break message are sent to the system operator's message queue, QSYSOPR.

message-queue-name: Specify the name of the message queue to which a reply to the break message is sent. Only a user or work station message queue can be specified.

CCSID

Specifies the coded character set identifier (CCSID) that the supplied message text is in. The text specified by the MSG parameter is assumed to be in the CCSID supplied by this parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center.

***JOB:** The message text is assumed to be in the CCSID of the job running this command.

***HEX:** The message text is not converted. CCSID 65535 is used.

coded-character-set-identifier: Specify a valid CCSID in which you want your message text to be considered in. Valid values range from 1 through 65535. See the Globalization topic in the Information Center for a list of valid CCSID values. This command will validate the CCSID.

Examples for SNDBRKMSG

Example 1: Sending a Message

```
SNDBRKMSG   MSG('The inventory application
  shuts down at 4:00 pm today.')
```

This command sends the message 'The inventory application shuts down at 4:00 pm today.' to all work station message queues. If the work station is signed on, the message will be delivered in break mode regardless of the delivery attribute setting of those message queues. The message is also added to the work station message queues of those work stations that are not signed on.

Example 2: Sending an Immediate Message

```
SNDBRKMSG   MSG('Your printed output is ready.')
  TOMSGQ(GEORGEMSGQ)
```

This example shows a typical use of the SNDBRKMSG command by the system operator to send an immediate message to a work station user.

Error messages for SNDBRKMSG

*ESCAPE Messages

CPF2428
Message queue parameter is not valid.

CPF2469
Error occurred when sending message &1.

CPF247E
CCSID &1 is not valid.

CPF9838
User profile storage limit exceeded.

SNDEMLIGC (Send DBCS 3270PC Emulation Code) Command Description

SNDEMLIGC Command syntax diagram

Purpose

The Send Double-Byte Character Set (DBCS) 3270PC Emulation Code (SNDEMLIGC) command sends a list of instructions to the system to download Personal System/55 (PS/55) Personal Computer code from an iSeries 400 to a PS/55 Personal Computer. Downloading is required if the user receives message CPF8557 while running the Start 3270 Device Emulation command (STREML3270) with *YES specified for the IGCEMLPC parameter.

The downloaded code is used to do SNA DBCS 3270PC emulation using the PS/55 Personal Computer. Emulation of SNA DBCS 3270PC display stations and printers connected to a Systems Network Architecture (SNA) 3274 is done from the downloaded code. SNA DBCS 3270PC emulation is called by the Start 3270 Display Emulation (STREML3270) command. The STREML3270 command must specify IGCEMLPC(*YES) requesting SNA DBCS 3270PC emulation. 3270 Display Station Passthru is running on the iSeries 400. Additional information on SNA DBCS 3270PC emulation is in the 3270 Device Emulation Support book, associated with the national language configured for your PS/55.

Optional Parameter

FROMFILE

Specifies the qualified name of the 3270PC file being downloaded to the PS/55.

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.

QAPJJPNB: The 3270PC file is downloaded to a Japanese PS/55 with an 8086 or 80286 processor.

Other IBM-supplied 3270 PC files that can be downloaded are:

QAPJJPNE: For a Japanese PS/55 with an 80386 processor.

QAPJJPNV: For a Japanese PS/55 with DOS J5.0.

QAPJKORB: For a Korean PS/55 with an 8086 or 80286 processor.

QAPJKORE: For a Korean PS/55 with an 80386 processor.

QAPJTCHB: For a Traditional Chinese PS/55 with an 8086 or 80286 processor.

QAPJTCHE: For a Traditional Chinese PS/55 with an 80386 processor.

3270PC-file-name: Specify the name of the 3270 PC file to be downloaded to your personal computer.

Example for SNDEMLIGC

```
SNDEMLIGC  FROMFILE(QAPJJPNE)
```

This command informs the user of the steps to be taken to download the 3270PC file QAPJJPNE to a Japanese PS/55 with an 80386 processor.

SNDDST (Send Distribution) Command Description

SNDDST Command syntax diagram

Purpose

The Send Distribution (SNDDST) command allows a user to send distribution to another user, to a list of users, or to a distribution list.

Restrictions:

1. A current user of this command working on behalf of another user must be granted authority to work on the behalf of the other user by means of the Grant User Permission (GRTUSRPMN) command.
2. Personal distribution cannot be received by a requester working on behalf of another user.
3. The DOCD, SUBJECT, and DSTD parameters are not updated from the SNDDST command to the actual mail log when a document (*DOC is specified for the TYPE parameter) is sent.

Required Parameters

TYPE Specifies the type of information sent and the valid values on this command. Dependent parameter codes for this parameter (see syntax diagram) determine which parameters are valid. If the dependent parameter code is listed in the syntax diagram, the parameter is valid when that type is specified.

***LMSG:** The text specified on the Long Message prompt (LONGMSG parameter) is sent as an final-form text document (FFTDCA) note. LONGMSG allows up to 5000 characters, which is about one page of text. The "details" parameters (Subject, Reference, Author, Due Date, etc.) are supported with TYPE(*LMSG).

***MSG:** Only the message specified in the MSG parameter is sent.

***DOC:** The document specified in the DOC or DOCID parameter is sent. The user must have authority for the document before it can be sent.

***FILE:** The database file specified in the DOCFILE and DOCMBR parameters is sent. The database file is sent without any changes. The user must have authority for the database file before it can be sent.

***IDP:** The interchange document profile (IDP) specified in the IDPFILE and the IDPMBR parameters, or the document profile built from the keywords DOCD, AUTHOR, DOCCLS, KWD, SUBJECT, REFERENCE, STATUS, PROJECT, ACTDATE, CMPDATE, CRTDATE, DOCDATE, FILCAB, CPYLST, and/or EXPDATE is used. The profile can be a profile of a printed document.

***DSTID:** The mail entry identified by the distribution ID is distributed. The distribution ID is referred to as the distribution document name.

TOUSRID

Specifies the user ID and address of one or more users to whom the distribution is being sent, or the distribution list name of one or more distribution lists containing the user ID and address of one or more users to whom the distribution is being sent. A combination of user IDs and distribution lists can be used on the same command. Up to 300 user IDs and addresses can be specified.

If no user ID and address is specified for the TOUSRID parameter, an internet address must be specified on the TOINTNET parameter.

Element 1: User ID or List ID

user-ID: Specify the user ID of the user to whom the distribution is sent.

list-ID: Specify the distribution list of users to whom the distribution is sent.

Element 2: User Address or List Qualifier

user-address: Specify the user address of the user (specified in element 1) to whom the distribution is sent.

list-qualifier: Specify the distribution list qualifier of users (specified in element 1) to whom the distribution is sent.

Element 3: Recipient Type

***PRI:** The user or distribution list is the primary recipient of the distribution.

***CC:** The user or distribution list is receiving a copy of the distribution sent to the primary recipient. This copy recipient is identified as a receiver on the distribution.

***BCC:** The user or distribution list is receiving a copy of the distribution sent to the primary recipient. However, this copy recipient is not identified as a receiver on the distribution.

DSTD Specifies the description of the distribution. A maximum of 44 characters can be specified.

TOINTNET

Specifies the internet address of one or more persons or organizations to whom the distribution is to be sent. If TOINTNET(*NONE) is specified, at least one user ID and address must be specified on the TOUSRID parameter. Up to 300 internet addresses can be specified.

The TOINTNET parameter is not allowed when the TYPE keyword is *MSG.

Single value

***NONE:** No internet address is specified.

Element 1: Internet Address

internet-address: Specify the internet address of a person or organization to whom the distribution is being sent. Up to 253 characters can be specified for each address.

Element 2: Recipient Type

***PRI:** The internet address is the primary recipient of the distribution.

***CC:** The internet address will receive a copy of the distribution sent to the primary recipient.

***BCC:** The internet address will receive a copy of the distribution sent to the primary recipient. However, this copy recipient is not identified as a receiver on the distribution.

Optional Parameters

MSG Specifies a short message.

***NONE:** No message is sent with the distribution.

***DSTIDMSG:** The same message in the distribution document specified on the DSTID parameter is sent with the distribution.

'message-text': Specify the message to send to the users. Up to 256 characters enclosed in apostrophes can be specified.

LONGMSG
Specifies the text to send to the recipients as an final-form text document (FFTDCA).

***NONE:** No text is sent. This is the default and required when TYPE is not *LMSG.

'long-message-text': Specify the text (5000 characters maximum) to send to the recipients. The text will be put into an final-form text document (FFTDCA) that will be sent as a note.

The text must be formatted by using the following controls within the text:

- :/N - Will cause a new line (Carrier return).
- :/P - Will cause a new paragraph. (New line plus a blank line)
  (Carrier return and Required carrier return)

The document is setup with 10 characters to the inch, the font ID is 11, the left margin at position 1, and the right margin at position 75. If :/N is not used to start a new line, the text will continue past the right margin and will not be viewable by some E-mail clients. Blanks are not removed from the text, but left in place for indentation and alignment. No extra text (ie. Recipient List, Date/Time, Subject, or Sender) is put into the document.

Technical note to programmers: Any of the FFTDCA controls (in hexadecimal) can be inserted in the text to provide more control of the format if the controls above are not enough.

DOC Specifies the name of the document being sent.

***DOCID:** The document being sent is identified by the library-assigned document name.

document-name: Specify the user-assigned name of the document being sent. Up to 12 characters can be specified.

FLR Specifies the name of the folder that contains the document. This is the user-assigned name given to the folder when it is created. If the DOC parameter is specified, the FLR parameter must also be specified.

***NONE:** No folder is specified when the document is identified by the DOCID parameter.

folder-name: Specify the name of the folder that contains the document being sent. A folder name can consist of a series of folder names (FLRA/FLRB/and so on) if the document being sent is located in a folder contained in another folder. Up to 63 characters can be specified.

DOCID

Specifies the library-assigned name of the document being sent. This is the name assigned to the document by the system when it was created. The library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command or the library-assigned document name is returned in a completion code when using the File Document (FILDOC) command. Library-assigned document names are 24 characters in length with the following format: YYYYMMDDHHMNSSHSSNSNSNSN where:

YYYY = year

MM = month

DD = day

HH = hour

MN = minute

SS = second

HS = hundredth of a second

SNSNSNSN = system name

***NONE:** No library-assigned document name is required when the document is identified by the DOC parameter.

'library-assigned-document-name': Specify the library-assigned document name being sent.

DOCFILE

Specifies the name of the database file that contains the document data to be sent. The database file is a user-defined file, or it is the output file specified by RCVDST(OUTFILE) or RTVDOC(OUTFILE). If an output file is specified, only the data portion of the document data record is read from the output file and the prefix is removed from the document data record. More information on defining the format of database files (OUTFILES) is in the Database Programming topic in the Information Center.

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.

DOCMBR

Specifies the document database file member to be sent.

***FIRST:** The first member in creation order in the database file is sent.

member-name: Specify the name of the database file member to be sent.

USRID

Specifies the user ID and address of the user for whom the request is made.

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

Element 1: User ID

user-ID: Specify the user ID of the user to whom the distribution is sent.

Element 2: User Address

user-address: Specify the user address of the user to whom the distribution is sent.

DSTID Specifies the unique distribution ID of the distribution. The ID is assigned to the distribution by the system that originated it.

Distribution IDs can be found by using the Query Distribution (QRYDST) command. IDs are also returned from the Send Distribution (SNDDST) command.

***NONE:** No distribution ID is sent.

'distribution-ID': Specify a distribution ID composed of the second part of the sender's user ID (padded on the right with blanks to 8 characters), the first part of the sender's user ID (padded on the right with blanks to 8 characters), and a 4-digit zoned sequence number with leading zeros. For example, *NEWYORK SMITH   0204*. The distribution ID is entered this way because a blank character is valid with a user ID address. This parameter is required when TYPE(*DSTID) is specified.

DSTIDEXN

Specifies the extension of the distribution ID if one is specified on the DSTID parameter. This extension uniquely identifies a duplicate distribution. This extension is a 2-digit extension ranging from 00 through 99. For incoming distributions, this extension ranges from 01 through 99. For confirmation of delivery distributions, this extension must be 00. 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 two distributions with the same distribution ID. To uniquely distinguish the two distributions, an extension is added to each distribution. For example, one distribution is identified by distribution ID and extension *NEWYORK SMITH 020401* and the other one by *NEWYORK SMITH 020402*. If there are no duplicate IDs, the extension defaults to 01. These extensions map one-to-one with the distribution ID that is specified on the DSTID parameter.

***NONE:** There is no duplicate distribution. This value is equivalent to a 01 extension.

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

DOCTYPE

Specifies the type of document being distributed. This identifier is used by the system to determine whether it can correctly handle the data stream.

***DFT:** The system creates the right document type identifier based upon the source of the data.

Document Type	Description
2	Final form text document
3	5520 revisable form text document
4	Word processing EBCDIC
5	Word processing information file
6	Image-data subset document
7	3730 text data stream
8	DIA document library descriptor document
9	3732 display document data stream
10	DIA display document data stream
11	Revisable form text document
12	1403 compatible data stream with variable length, unblocked records
13	Digitized ADS audio
14	IBM PC data file
15	Hard copy of document
223	Data file

If TYPE(*IDP) is specified, the type is 15 (printed document).

If TYPE(*FILE) is specified and the file is an output file created by the RTVDOC command (OUTFILE parameter), then the document type is taken from this file. If this file does not contain a document type or the value in the file for the document type is 0, an error message is sent.

If the file is a user-defined file, the document type is 223 (data file).

If TYPE(*DOC) is specified, the document type of the document sent is used. If TYPE(*DSTID) is specified, the document type is the same document type that is associated with the distribution.

***FFT:** The document is a final-form document. This type of document is intended to be shown and printed, but not edited, by the receiver.

***RFT:** The document is a revisable-form text. This type of document can be viewed, printed, and edited by the receiver.

document-type-number: Specify a value ranging from 2 through 65535. The numbers ranging from 2 through 32767 are controlled by registering them with the IBM Document Interchange Architecture and are used for IBM-defined document types. The numbers ranging from 32768 through 65535 are not registered with IBM and can be used for non-IBM defined document types. The meaning of these document types must be stated as values of the SYSCOD parameter.

SNDFMT

Allows the user to specify the format of the document being sent.

***NOCHG:** The document is sent in the current format.

***NOTE:** The document is sent in a final form text document content architecture (FFTDCA) data stream as a note.

***FINALFORM:** The document is sent in FFTDCA.

SYSCOD

Specifies text used with the document type parameter to help identify the type of document being sent. The receiver of the data stream determines the document data stream and processing requirements to edit, view, print or change the document.

***DFT:** The system supplies a default system code. If the DOCTYPE value is within the range 2 through 32767, the default is 'IBM iSeries 400 CL'. If the DOCTYPE is within the range 32768 through 65535, a system code must be specified.

'system-code': Specify the text that helps uniquely identify the type of document being sent. Up to 13 characters can be specified.

SENSITIV

Specifies the level of sensitivity defined by the X.400 standard. The four levels include normal, personal, private and company confidential. Private distributions cannot be viewed by a user working on behalf of another user.

***NONE:** The distribution has no sensitivity restrictions.

***PERSONAL:** The distribution is sent to the recipient as an individual.

***PRIVATE:** The distribution contains information that should be accessed only by the recipient.

***CONFIDENTIAL:** The distribution contains information that should be handled according to company procedures.

PERSONAL

Specifies whether the document distribution is private or not. This parameter is replaced by SENSITIV but the PERSONAL parameter can still be used. However, because this parameter may be removed in a later release, whenever possible use the SENSITIV parameter.

If PERSONAL(*YES) is used, the SENSITIV parameter must be omitted or it must be with the value SENSITIV(*NONE). If the command is prompted without this parameter specified, this parameter is not displayed.

***NO:** Only the owner and users that have authorization to the distribution document can get access to distributions that are not sensitive. Users authorized to work on behalf of other users who have access to the distribution can access documents that are not sensitive. This value will map to SENSITIV(*NONE).

***YES:** Only the owner can get access to private distribution documents. Users authorized to work on behalf of other users who have access to the distribution document cannot get access to the distribution. This value will map to SENSITIV(*PRIVATE).

IMPORTANCE

Specify low, normal or high importance. This is an indication to the recipient of the content importance of the distribution. It is different from priority which relates to the speed with which the distribution was sent.

***NORMAL:** A distribution of normal importance.

***HIGH:** A very important distribution.

***LOW:** A low importance distribution.

PTY Specifies whether the distribution is sent using low priority, normal priority or high priority. For distributions to remote receivers, the priority determines which 'SNADS Next System Queue' is used. Normal priority distributions use the normal next system queue. Low priority distributions use the normal next system queue with the lowest priority available within that queue. High priority uses the priority next system queue. The difference between low, high and normal priority depends on information, specified by the user, on the Configuration Distribution Services (CFGDSTSRV) command. The handling of priority distributions by other office system nodes can vary, but generally the high priority distributions take the faster path when there is a choice of paths. For distribution to local receivers, the priority determines whether a message is sent to the receiver's message queue to notify the receiver of the distribution. No message is sent for normal distributions.

***NORMAL:** Normal priority is used.

***HIGH:** High priority is used.

***LOW:** Low priority is used.

CFMDEL

Specifies whether the sender wants a confirmation of delivery notification when each receiver receives the distribution. If confirmation of delivery is not requested, the sender is still informed if the distribution is not delivered because of system failures, routing failures, or because it has an invalid user ID. If confirmation of delivery is requested, the sender is informed when the receiver either receives the distribution or deletes the distribution before receiving it. To get this information, the sender must do a Query Distribution (QRYDST) command with the OPTION(*OUT) specified.

***NO:** Request for confirmation of delivery is not specified.

***YES:** Request for confirmation of delivery is specified.

IDPFILE

Specifies the name of a database file that contains the interchange document profile (IDP) data. The interchange document profile is either a user-defined file or the output file specified on the OUTFILE parameter of the Receive Distribution (RCVDST) or Retrieve Document (RTVDOC) commands. If an output file is specified, only the data portion of the document profile record is read from the output file and the prefix is removed from the IDP data.

If this parameter is specified, the remaining parameters on this command are ignored except for the DOCCHRID and CMDCHRID parameters.

***NONE:** The interchange document profile is not in a database file. The information is provided by other parameters on this command.

***DSTIDIDP:** The interchange document profile (IDP) information associated with the distribution document is used. This is valid only when TYPE(*DSTID) is specified.

***DOCFILE:** The interchange document profile is in the same file as the document. A member specified in the IDPMBR parameter is ignored.

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 contains the interchange document profile. More information on defining the format of database files (output files) is in the Database Programming topic in the Information Center.

IDPMBR
Specifies the interchange document file member name to send. This parameter is valid only when IDPFILE(database-file-name) is specified.

***FIRST:** The first member (based on creation date) in the database file is sent.

member-name: Specify the name of the database file member to be sent.

DOCD Specifies a description for the document being distributed. This is the Document Interchange Architecture IDP document name field.

***DFT:** The system creates a document description for database files. The default is library-name/file-name/member-name. For a description of a printed document (TYPE(*IDP)), the default is the distribution description. For a description of a distribution document (TYPE(*DSTID)), the default is the document description associated with the distribution.

***DSTD:** The distribution description is used for the document name.

'document-description': Specify up to 44 characters for the description of the document.

AUTHOR
Specifies the authors of the document.

***NONE:** No author is identified for the document.

***USRID:** The user ID and address of the user who is requesting the distribution is used as the author's name.

'document-author': Specify the name of the authors. Up to 50 authors can be specified.

DOCCLS
Specifies the class associated with this document, such as MEMO, FORM, or SHEET.

***NONE:** No class is assigned to the document.

'document-class': Specify the document class. Up to 16 characters can be specified.

KWD Specifies the keywords that can be used to describe the document.

***NONE:** No keywords are defined for this document.

'document-keyword': Specify the keywords to use to describe the document. Up to 50 keywords can be specified and each keyword can have up to 60 characters.

SUBJECT
Specify the subjects of the document.

***NONE:** No subjects are defined for the document.

***DOCD:** The document description is used as the subject for the document.

'document-subject': Specify the subjects of the document. Up to 50 subjects can be specified and each subject can have up to 60 characters of text.

DOCDATE
Specifies the date the user wants to assign to the document.

***NONE:** No date is assigned to the document.

***CURRENT:** The current date is used.

document-date: Specify the value used as the document date. The date must be in the format specified by the system value QDATFMT.

FILCAB
Specifies the location of the document. This parameter is intended for printed documents. The interchange document profile, which refers to the printed document, is distributed. This parameter is required if TYPE(*IDP) and IDPFILE(*NONE) are specified.

***NONE:** No filing cabinet reference is defined for this document.

'filing-cabinet-reference': Specify the text that describes where the printed document is located. Up to 60 characters can be specified.

CPYLST

Specifies names of the users who receive this distribution.

***NONE:** No copy list is included for this distribution.

'recipient-list': Specify the names of the users who receive the document. Up to 50 names can be specified and each name can have up to 60 characters.

EXPDATE

Specifies the date on which the document is no longer needed.

***NONE:** No expiration date is specified.

expiration-date: Specify the value to use as the expiration date. The date must be specified in the format specified by the system value QDATFMT.

CRTDATE

Specifies the creation date of the object.

***NONE:** A document creation date is not specified.

***CURRENT:** The current date is used.

create-date: Specify the document creation date. The date must be in the format specified by the system value QDATFMT.

REFERENCE

Specifies a reference associated with the document.

***NONE:** No reference field is included in the interchange document profile.

'reference': Specify the text that describes the reference associated with the document. Up to 60 characters can be used.

ACTDATE

Specifies the due date for the requested action. If no action due date is specified and a reply request date is specified the action due date is set from the reply request date.

***NONE:** No action due date is specified.

***CURRENT:** The current date is used.

action-due-date: Specify the value used as the action due date. The date must be specified in the format specified by the system value QDATFMT.

RPYDATE

Specifies the date and time a reply to the distribution is requested. This applies only to primary recipients.

Element 1: Reply by Date

***NONE:** No reply is required.

***CURRENT:** The current date is used. The time will default to *ENDOFDAY and will be set to 23:59:59.

***ANY:** A reply is requested but no date and time are specified.

reply-by-date: Specify the value used as the reply requested date. The date must be specified in the format specified by the system value QDATFMT.

Element 2: Reply by Time

***ENDOFDAY:** A reply is requested by the end of the specified date. The time is set to 23:59:59.

reply-by-time: Specify the value used as the reply requested time.

The time is specified in 24-hour format and can be specified with or without a time separator.

Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh=hours, mm=minutes, and ss=seconds. Valid values for hh range from 00 to 23. Valid values for mm and ss range from 00 to 59.

With a time separator, specify a string of 5 or 8 digits (hours and minutes, or hours, minutes, and seconds). The time separator specified by the system value QTIMSEP is used to separate the hours, minutes, and seconds. If the command is entered from the command line, the string must be entered in apostrophes. If a time separator other than the separator used for your job is used, the command fails. Valid values for hh range from 00 to 23. Valid values for mm and ss range from 00 to 59.

STATUS

Specifies the user-defined status (In Process, Pending Approval, or Retired).

***NONE:** No status is included in the interchange document profile.

'status-of-document': Specify text that describes the status of the document. Up to 20 characters can be specified.

CMPDATE

Specifies the completion date for the requested action.

***NONE:** No completion date is specified.

***CURRENT:** The current date is used.

date-complete: Specify the completion date. The date must be in the format specified by the system value QDATFMT.

PROJECT

Specifies the project with which the document is associated.

***NONE:** No project field is included in the interchange document profile.

'project': Specify text that describes the project of the document. Up to 10 characters can be specified.

DOCCHRID

Specifies the character identifier (graphic character set and code page) for the data being distributed. The character identifier is related to the display device that was used to create the data being distributed. More information about CHRID processing is in the Database Programming topic in the Information Center.

***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 to use for creating the data being distributed.

Element 2: Code Page

code-page: Specify the code page value used to create the command parameters. Valid values range from 1 through 999.

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: The CMDCHRID parameter applies to the following parameters and means in some cases that the data is translated to a code page and character set that is interchangeable with other IBM office products. The interchangeable character set and code page is '697 500' except for USRID, TOUSRID, and DSTID which are '930 500'. In other cases, the code page and character set are attached to the field and sent with it to allow the receiving terminal to correctly print and display the field.

Parameter	Translate
TOUSRID	Translated
USRID	Translated
DSTID	Translated
SYSCOD	Translated
MSG	Translated
DOCD	Passed
DSTD	Translated
AUTHOR	Passed
DOCCLS	Passed
KWD	Passed
SUBJECT	Passed
FILCAB	Passed
CPYLST	Passed
REFERENCE	Passed
STATUS	Passed
PROJECT	Passed
AUTUSR	Translated

The value translates the USRID and DSTID parameter to 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 to use for creating the command parameter.

Element 2: Code Page

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

DOCLANGID

Specifies the language identifier being placed in this distribution document's interchange document profile (IDP). This parameter is ignored if the IDPFILE parameter is specified.

***JOB:** The language identifier specified for the job in which this command is entered is used.

language-identifier: Specify the language identifier. More information on valid language identifiers is in the Globalization topic in the Information Center. More information on this parameter is in commonly used parameters.

DOCCNTRYID

Specifies the country or region identifier being placed in this distribution document's interchange document profile (IDP). This parameter is ignored if the IDPFILE parameter is specified.

***JOB:** The country or region identifier specified for the job in which this command is entered is used.

country-identifier: Specify an ISO 3166 Alpha-2 code from the country or region code table. More information on this parameter is in commonly used parameters.

ALWALTRCP

Specify if the distribution can be delivered to an alternate recipient, determined by the receiving system. This function is used only by X.400 network systems. This function provides the ability to direct certain mail to a specific user.

The default of this field is *YES. However, because the user might change the sensitivity field to private, but might not change the alternate recipient field to no, he may send a private item to an alternate recipient. No error message will be generated for this condition.

***YES:** Specifies the distribution can be delivered to an alternate recipient.

***NO:** Specifies the distribution cannot be delivered to an alternate recipient.

DISCLOSE

Specifies whether or not each recipient gets a list of the other recipients.

***YES:** Disclose recipients.

***NO:** Do not disclose recipients.

ALWX400CNV

Specifies whether an X.400 conversion is allowed on the distribution being submitted. In certain circumstances, the end user knows the capabilities of the recipient and may indicate that the distribution is not to be converted. The system administrator can define whether or not conversion is to take place. This field allows the user to override that setting on a message by message basis. For example, if the gateway allows conversion, the user could specify that conversion is to be prohibited. This will take precedence over the gateway setting. The distribution would then not be converted.

Note: This field is valid only for X.400 support. This field will not affect iSeries 400 data stream transformations, such as RFT to FFTDCA.

***YES:** Conversion may be performed on the distribution by the receiving system.

***NO:** No conversion is permitted on the distribution by the receiving system.

AUTUSR

Specifies the user ID and address of the user that authorized the content of this distribution. The authorizing user will receive a copy of the distribution.

***NONE:** No authorizing user.

Element 1: User ID

user-ID: Specify the user ID of the user from whom the distribution is authorized.

Element 2: User Address

user-address: Specify the user address of the user from whom the distribution is authorized.

DSTEXPDATE
Specifies the date and time at which the distribution is no longer needed in the mail log.

Element 1: Distribution Expiration Date

***NONE:** This distribution has no expiration date.

distribution-expiration-date: Specify the value to use as the expiration date. The date must be in the format specified by the system value QDATFMT.

Element 2: Distribution Expiration Time

ENDOFDAY: An expiration time is requested by the end of the specified date. The time will be set to 23:59:59.

distribution-expiration-time: Specify the value as the expiration time.

The time is specified in a 24-hour format and can be specified with or without a time separator.

Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh=hours, mm=minutes, and ss=seconds. Valid values for hh range from 00 to 23. Valid values for mm and ss range from 00 to 59.

With a time separator, specify a string of 5 or 8 digits (hours and minutes, or hours, minutes, and seconds). The time separator specified by the system value QTIMSEP is used to separate the hours, minutes, and seconds. If the command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator used for your job is used, the command will fail. Valid values for hh range from 00 to 23. Valid values for mm and ss range from 00 to 59.

Examples for SNDDST

Example 1: Sending a Distribution on Behalf of Another User

```
SNDDST  TYPE(*FILE)  TOUSRID((JACKSON RCH38DB))
  DOCTYPE(20000)  SYSCOD(BRANDX)
  DOCFILE(DEPT46ELIB/XTEXT)  DOCMBR(GOLD1IPFS)
  PTY(*HIGH) USRID(JACOBSON RCH38NBS)
  DSTD('IPFS FOR GOLD1 PROJECT')
  MSG('review and update section 1.2.4.
  return for final printing by 2/15/89')
  CFMDEL(*YES)
```

This command sends a distribution that is being sent by someone (such as a secretary) who is authorized to work on behalf of JACOBSON. The document being sent is a BRANDX text document that is sent to another user who also has the BRANDX text processor.

Example 2: Sending a Mail Log Entry

```
SNDDST  TYPE(*DSTID)  DSTID('NEWYORK SMITH 0204')
  DSTIDEXN(02)  TOUSRID((JACKSON RCH38DB))
  MSG(*DSTIDMSG)  CFMDEL(*YES)
```

This command sends a mail log entry that is identified by the distribution document name *NEWYORK SMITH 0204* that is distributed to user JACKSON at address RCH38DB. The message in the distribution document is distributed with the distribution.

Error messages for SNDDST

***ESCAPE Messages**

CPF8A87
Document name &2 not correct.

CPF8A97
Folder name &1 not correct.

CPF89AA
*FINALFORM for send format (SNDFMT) valid only for *DOC for type (TYPE).

CPF89AB
Specify *IDP or *MSG for type (TYPE) only with *NOCHG for send format (SNDFMT).

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

CPF900C
Sign on and verify of user failed.

CPF901A
Send distribution request failed.

CPF903D
Incorrect document identifier specified.

CPF905C
Error occurred trying to find a translation table.

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

CPF9845
Error occurred while opening file &1.

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

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

SNDDSTQ (Send Distribution Queue) Command Description

SNDDSTQ Command syntax diagram

Purpose

The Send Distribution Queue (SNDDSTQ) command is used:

- To send a distribution queue's entries when the distribution queue is configured to be manually started but no operator is available.
- To override any distribution queue scheduling attributes and begin sending a queue's entries immediately.
- To restart a SNADS sender job that failed abnormally.

The SNDDSTQ command is primarily intended for use in a batch CL program. The SNDDSTQ command enables the same functions as option 2 (Send distribution queue) on the Work with Distribution Queue (WRKDSTQ) command main list panel. The SNDDSTQ command allows the functions to be started from a batch job instead of interactively.

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

Restrictions:

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 sent. Specify the name of the distribution queue. The queue specified must have been previously configured using the Configure Distribution Services (CFGDSTSRV) command or the Add Distribution Queue (ADDDSTQ) command.

PTY Specifies whether the normal or high priority portion of the specified queue is sent.

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

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

Examples for SNDDSTQ

Example 1: Sending Distributions with Normal Priority

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

This command sends distributions from the normal priority portion of the CHICAGO distribution queue.

Example 2: Sending Distributions with High Priority

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

This command sends distributions from the high priority portion of the ATLANTA distribution queue.

Error messages for SNDDSTQ

*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.

SNDF (Send File) Command Description

SNDF Command syntax diagram

Purpose

The Send File (SNDF) command is used by a control language (CL) program to send a record to a display device that is being used by an interactive user. The device can be any display station including the console. The command sends the data from the program's CL variables to the display's device file in the specified record format. These variables were automatically declared in the program (one for each field in the record format) when the CL source program was compiled and a Declare File (DCLF) command was processed as part of the source.

Of the record formats specified in the DCLF command, only one can be specified in each SNDF command. If the device file has not been opened, it is opened by this command. The file and record format specified in this command can be overridden by an Override with Display File (OVRDSPF) command if it is entered before the file is opened. However, care should be taken that the fields in the overriding record format correspond to the CL variables declared in the program.

Restrictions:

1. This command is valid only within a CL program.
2. This command is valid only for display files.
3. This command cannot be used with database files.

Optional Parameters

DEV Specifies the name of the display device to which the data in the CL variables for the specified record format is sent.

***FILE:** The program's data is sent to the device associated with the device file that was declared in the FILE parameter of the DCLF command. If more than one device name is specified in the device file, *FILE cannot be specified.

device-name: Specify the name of the device or the name of the CL variable that contains the name of the device to which the program's data is sent.

RCDFMT

Specifies the name of the record format that is used to send data to the file. The format contains all the fields in the record. This parameter must be coded with a record format name if there is more than one record format name in the device file; *FILE cannot be coded if there is more than one. If the record format contains the INVITE DDS keyword (optioned on), the SNDF functions as if SNDRCVF WAIT(*NO) had been coded.

***FILE:** There is only one record format in the device file; that is the format in which the program's data is sent to the file.

record-format-name: Specify the name of the record format in which the program's data is sent to the file. A CL variable cannot be used here to specify the record format name.

Example for SNDF

```
DCLF   FILE(MENU1)
  *
  *
  *
SNDF
```

The record format in the device file MENU1 is sent to the device specified in the file. There is only one record format in the file.

```
DCLF   FILE(SCREEN1)  RCDFMT(REC1 REC2)
  *
  *
  *
SNDF   DEV(DISP3)  RCDFMT(REC1)
```

The device file named SCREEN1 causes the display station named DISP3 to display the data sent by the CL program to the user. The data is shown in the format specified by the REC1 record format.

```
DCLF   SCREEN1  (REC1 REC2)
  *
  *
  *
SNDF   *N  REC2
```

The device file named SCREEN1 sends data to the device named in the same device file. *N is the null value for the DEV parameter, coded positionally, that defaults to *FILE. The data is presented to the user in the format specified by REC2.

Error messages for SNDF

*ESCAPE Messages

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

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

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

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

CPF0887
Data available from previous input request.

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

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

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

SNDFNCIMG (Send Finance Diskette Image) Command Description

SNDFNCIMG Command syntax diagram

Purpose

The Send Finance Diskette Image (SNDFNCIMG) command allows the user to send an operational diskette image to a 4701 finance controller, which takes the image and builds a new operational diskette that is used to start the controller.

The user must first create the diskette image on a host system (for example, System/370* system) by using the Host Diskette Image Create package. The user must also provide a host program to block the image into a basic exchange file format and then provide a way of sending the blocked image to an iSeries 400 and placing it in a file. This command rebuilds the original image and sends it to the controller, which writes the image on a blank diskette and builds an operating diskette containing system code, configuration files, and application programs. The diskette can then be used to start the controller.

Restriction: The user must have QSECOFR authority to use this command.

Required Parameters

FILE Specifies the qualified file name where the diskette image is located.

The file being sent must be a physical file with a record length of 80 bytes. Save files, logical files and device files are not allowed. Overrides to the specified 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.

file-name: Specify the name of the file where the diskette image is located.

RMTLOCNAME
Specifies the remote location name of a finance device with TYPE(*FNCICF). This device must be attached to a 4701 finance controller with an 8-inch diskette drive or a 3601 controller configured as a 4701 controller. The local location address (LOCADR) of the device *must* be 01.

Optional Parameter

MBR Specifies the member in the file that contains the diskette image that is blocked into a basic exchange format.

***FIRST:** The first member in the database file is used.

member-name: Specify the name of the member that contains the image to be blocked.

Example for SNDFNCIMG

```
SNDFNCIMG  FILE(IMAGEFILE)  MBR(OTSIMAGE)
  RMTLOCNAME(SYSMON1)
```

This command builds the diskette image from member OTSIMAGE in file IMAGEFILE and then sends it to remote location SYSMON1.

Error messages for SNDFNCIMG

***ESCAPE Messages**

CPF87A9
Device with RMTLOCNAME &1 not found.

SNDJRNE (Send Journal Entry) Command Description

SNDJRNE Command syntax diagram

Purpose

The Send Journal Entry (SNDJRNE) command is used to write a single journal entry to a specific journal. The entry can contain any information. The user may assign an entry type to the journal entry and may also associate the journal entry with a specified journaled object. »

If the journal currently has a state of *STANDBY, then the journal entry will not be deposited unless OVRSTATE(*STANDBY) is specified. «

The journal code for the entry is ’U’, which indicates a user-specified journal entry.

Note: The Send Journal Entry (QJOSJRNE) Application Programming Interface (API) can also be used to write a user-specified journal entry to a specific journal. Using this API may improve performance and can provide additional function that is not available with this command. For more information, see the Application Program Interfaces (APIs) topic in the Information Center.

Restrictions:

1. If an object other than a file is specified, it must be currently journaled to the specified journal.

   If a file object is specified, it must either be currently journaled to the specified journal or it must have been last journaled to the specified journal.
2. The specified journal cannot be a remote journal.
3. The specified journal cannot have a journal state of *INACTIVE.

Required Parameter

JRN Specifies the name and library of the journal to contain the new journal entry.

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 and library of the journal to contain the new journal entry.

Optional Parameters

TYPE Specifies the journal entry type of this journal entry.

’00’: The journal entry type is a ’00’ (hex F0F0).

entry-type: Specify a 2-character value or hexadecimal value used for the journal entry type. This value must be greater than or equal to hex C000.

If a hexadecimal value is specified that does not represent characters, that value is not shown on the DSPJRN display or on the printout.

ENTDTA

Specifies the user-specified data that is placed in the variable portion of the journal entry.

***BLANK:** No user-specified data is placed in the journal entry.

’entry-specific-data’: Specify up to 3000 characters, enclosed in apostrophes.

FILE Specifies the qualified name of the database physical file and member with which this entry is associated.

If this parameter is specified, the parameters OBJ, OBJPATH or OBJFID cannot also be specified.

Element 1: Physical File

The possible Physical File values are:

***NONE:** There is no associated physical file for this entry.

file-name: Specify the name of the physical file with which this entry is associated.

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.

Element 2: Member

The possible Member values are:

***FIRST:** The entry is associated with the first member in the file.

member-name: Specify the name of the physical file member with which this entry is associated.

OBJ Specifies the qualified object name with which this entry is associated.

If this parameter is specified, the parameters FILE, OBJPATH or OBJFID cannot also be specified.

Element 1: Object

The possible Object values are:

***NONE:** There is no associated object for this entry.

object-name: Specify the name of the object with which this entry is associated.

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

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

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

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

Element 2: Object Type

The possible Object Type values are:

***FILE:** The entry is associated with a database file member.

***DTAARA:** The entry is associated with a data area.

***DTAQ:** The entry is associated with a data queue.

Element 3: Member

The possible Object Member values are:

***FIRST:** The entry is associated with the first member in the file.

member-name: Specify the name of the physical file member with which this entry is associated.

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

OBJPATH

Specifies the path-qualified object name with which this entry is associated. Only objects whose path name identifies an object of type *STMF, *DIR or *SYMLNK that is in the Root ('/'), QOpensys, and User-defined file systems are supported.

If this parameter is specified, the parameters FILE, OBJ or OBJFID cannot also be specified.

***NONE:** There is no associated object for this entry.

path-name: Specify the name of the object with which this entry is associated. Symbolic links within the path name will not be followed. For more information on specifying path names, refer to path names.

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

OBJFID

Specifies the file-identifier (FID) with which this entry is associated. FIDs are a unique identifier associated with integrated file system related objects. This field is input in Hexadecimal format. Only objects whose FID identifies an object of type *STMF, *DIR or *SYMLNK that is in the Root ('/'), QOpensys, and User-defined file systems are supported.

If this parameter is specified, the parameters FILE, OBJ or OBJPATH cannot also be specified.

***NONE:** There is no associated object for this entry.

file-identifier: Specify the FID for the object with which this entry is associated.

FORCE

Specifies whether the journal receiver is forced to auxiliary storage after the user entry is written to it.

***NO:** The journal receiver is not forced to auxiliary storage.

***YES:** The journal receiver is forced to auxiliary storage.

OVRSTATE

Specifies whether the journal entry will be deposited, overriding the current state of the journal.

***NONE:** None of the journal state values are overridden. That is,

- The journal entry is deposited if the journal state is *ACTIVE.
- The journal entry is not deposited and an error is sent if the journal state is *INACTIVE.
- The journal entry is not deposited and no error is sent if the journal state is *STANDBY.

***STANDBY:** The journal entry is deposited even if the journal state is *STANDBY.

Examples for SNDJRNE

Example 1: Forcing Journal Receivers to Auxiliary Storage

```
SNDJRNE   JRN(JRNLA)  TYPE(AB)
  ENTDTA('PROGRAM COMPLETE')
  FILE(MYLIB/ORDERENT MBR1)  FORCE(*YES)
```

If the journal currently has a journal state of *ACTIVE, this command places a journal entry of type AB (hex C1C2) with the journal entry data ’PROGRAM COMPLETE’ in the current journal receivers attached to journal JRNLA as found by using the library search list. The entry is associated with member MBR1 of file ORDERENT in library MYLIB. The journal receivers are forced to auxiliary storage after the entry has been placed in them.

Example 2: Sending a Journal Entry

```
SNDJRNE   JRN(JRNLA)  TYPE(x'C1F1')
          OVRSTATE(*STANDBY)
```

If the journal currently has a journal state of *STANDBY or *ACTIVE, this command places a journal entry of type ’A1’ (hex C1F1) with no journal entry data in the current journal receivers attached to journal JRNLA as found by using the library search list. The entry is not associated with any physical file member.

Error messages for SNDJRNE

***ESCAPE Messages**

CPFA0D4

File system error occurred.

CPF7002

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

CPF7003

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

CPF7004

Maximum number of objects journaled to journal &1.

CPF7007

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

CPF7035
Object &1 in &2 already known to journal.

CPF7037
Object not journaled to journal &3.

CPF70EF
Parameters cannot be used together &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.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

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

CPF9820
Not authorized to use library &1.

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

SNDLIC (Send License) Command Description

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

SNDLIC Command syntax diagram

Purpose

The Send License (SNDLIC) command provides the capability to send one or more license keys from the central site system to one or more managed systems. Only the license keys matching the system serial number will be added to the managed system license repository.

Notes:

1. A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.
2. All the existing records in the central site license repository containing the license key information for the specified product will be sent to the specified managed system or systems. Only those ones that match the system serial number are added to the managed system license repository.

Restrictions:

1. This command is shipped with public authority of *EXCLUDE.
2. If a node list (NODL) value is specified, the node list can only contain entries that have a value of *SNA for the address type.

3. The license key information for the specified product and managed system must exist in the central site license repository.

Required Parameters

PRDID

Specifies the 7-character identifier of the product for which the license key information will be distributed.

***ALL:** The license keys for all the products existing in the central site license repository will be distributed.

product-ID: Specifies the 7-character identifier of the product for which the license key information will be distributed.

LICTRM

Specifies the license term for which the licenses key information will be sent.

***ALL:** The product licenses for all license term found will be sent.

license-term: Specify the license term in one of the following formats:

Vx The authorized usage limit is valid for the entire version of the product or feature.

VxRy The authorized usage limit is valid for the entire release of the product or feature.

VxRyMz

The authorized usage limit is valid only for a modification of the product.

where x and y can be a number from 0 through 9, and z can be a number 0 through 9 or a letter A through Z.

NODL Node list object name that contains a list of systems which are the destinations for the request. Cannot be specified if control point names (CPNAME parameter) are also specified.

***NONE:** The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible library values are:

***LIBL:** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

***CURLIB:** The current library for the job is used to locate the node list object.

library-name: Only the library named in this keyword is searched.

node-list-name: Node list object name which contains the list of systems on which the command is to be performed.

CPNAME

The APPN control point names of the managed systems on which this activity is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

***NONE:** The systems on which this activity is to be performed are not identified individually. A node list must be specified.

***NETATR:** The network ID of the local system will be used. This useful when the node being specified is in the same network as the local system.

network-identifier: The APPN network identifier of the managed system on which the activity is to be performed, it is a character field of length 8.

control-point-name: The APPN control point name of the managed system on which the activity is to be performed, it is a character field of length 8.

Examples for SNDLIC

```
SNDLIC  PRDID(1ACCOUN)
  LICTRM(V4)
  CPNAME((*NETATR SYS1))
```

Send the licenses for 1ACCOUN product with license term V4, and add the licenses matching the system serial number to the repository of system SYS1.

Error messages for SNDLIC

***ESCAPE Messages**

None «

SNDMSG (Send Message) Command Description

SNDMSG Command syntax diagram

Purpose

The Send Message (SNDMSG) command is used by display station users to send immediate messages from their display stations to one or more message queues. An immediate message is a message that is not predefined and is not stored in a message file. The message can be sent to the system operator, to other display station users, to a user's message queue, to all currently active user's message queues, or to the system history log, QHST. The sender can require a reply from the message receiver. The primary users of this command are display station operators and the system operator.

Restrictions:

1. The user must have object operational and add authorities to the message queue.
2. The user of this command must have *USE authority for the specified message queues and for the libraries in which they are stored.
3. The SNDMSG command only allows a message of up to 512 characters of first-level message text to be sent.
4. This command can only send inquiry messages (specified by MSGTYPE(*INQ)) to one message queue or to two message queues if one of the queues is *HSTLOG.

Required Parameters

MSG Specifies the message text that is sent. A maximum of 512 characters can be specified. The message must be enclosed in apostrophes if it contains blanks or other special characters.

Coded Character Set Identifier (CCSID) Considerations

The text supplied on the MSG parameter is assumed to be in the CCSID of the job running this command unless a coded character set identifier is supplied in the CCSID parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center.

TOUSR

Specifies that the message is sent to the message queue specified in the user profile for the user named on this parameter. This parameter cannot be used if TOMSGQ is specified.

***SYSOPR:** The message is sent to the system operator's message queue (QSYS/QSYSOPR). Any message sent to QSYSOPR automatically has a copy of the message sent to QHST.

***REQUESTER:** The message is sent to the user profile's message queue for interactive jobs or to the system operator's message queue (QSYS/QSYSOPR) for batch jobs.

***ALLACT:** A copy of the message is sent to the user profile message queue of each user profile with an interactive job currently running. This value cannot be specified if the value *INQ is specified for the MSGTYPE parameter.

user-profile-name: Specify the user profile name of the user to whom the message is sent.

TOMSGQ

Specifies the qualified names of up to 50 message queues to which an informational message is sent. For an inquiry message, one message queue may be specified or two message queues may be specified if one of the queues is *HSTLOG. This parameter cannot be used if TOUSR is specified.

***SYSOPR:** The message is sent to the system operator's message queue, QSYS/QSYSOPR. Any message sent to QSYSOPR automatically has a copy of the message sent to QHST.

The name of the message 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.

message-queue-name: Specify the name of the message queue to which messages are sent.

***HSTLOG:** The message is sent to the system history log message queue, QSYS/QHST. If *HSTLOG is specified more than once, only one message will be sent to QSYS/QHST. If *HSTLOG is specified with QSYSOPR only one message is sent to QSYS/QHST.

Optional Parameters

MSGTYPE

Specifies the type of message being sent. Only informational and inquiry message types can be specified.

***INFO:** An informational message is sent.

***INQ:** An inquiry message is sent. The message queue receiving the message can reply to it. Inquiry messages are sent to only one message queue at a time or a second queue can be specified for TOMSGQ if the value is *HSTLOG.

RPYMSGQ

Specifies, only if an inquiry message is sent, the qualified name of the message queue to which a reply is sent.

***WRKSTN:** The reply to the message is sent to the display station message queue associated with the sender's display station.

Note: This value cannot be specified for batch jobs.

The name of the message 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.

message-queue-name: Specify the name of the message queue to which the reply is sent.

CCSID

Specifies the coded character set identifier (CCSID) that the supplied message text is in. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center.

***JOB:** The message text is assumed to be in the CCSID of the job running this command.

***HEX:** The message text is not converted. CCSID 65535 is used.

coded-character-set-identifier: Specify a valid CCSID in which you want your message text to be considered in. Valid values are between 1 and 65535. This command validates the CCSID.

Examples for SNDMSG

Example 1: Sending Message to User Message Queue

```
SNDMSG   MSG('Do you want to update INV now?')
  TOUSR(JONES)  MSGTYPE(*INQ)  RPYMSGQ(SMITH)
```

This command sends a message to the user message queue JONES. When the message is answered, the reply will be sent to the message queue SMITH.

Example 2: Sending Message to System's History Log

```
SNDMSG   MSG('Input errors on PAYROLL cost
  me 1 hour of run time.')  TOMSGQ(QHST)
```

This command is used by the system operator to send an informational message to the system's history log, QHST, through the log's message queue, which has the same name.

Example 3: Sending Message to System Operator

```
SNDMSG   MSG('Please make 2 copies of printer
  file LABORSTAT.')  TOMSGQ(QSYSOPR)
```

This example shows a typical use of the SNDMSG command by a display station user. The user is sending the message 'Please make 2 copies of printer file LABORSTAT.' to the system operator.

Example 4: Sending Message that Requires a Reply

```
SNDMSG   MSG ('How long will the display stations be
  online today?')  TOMSGQ(*SYSOPR)  MSGTYPE(*INQ)
```

This inquiry message from a display station user requires a reply. The system operator displays the message by using the DSPMSG command and enters the reply on the display. The reply is then sent to the display station user's work station message queue. The display station user enters another DSPMSG command to display the reply.

Error messages for SNDMSG

***ESCAPE Messages**

CPF2428
Message queue parameter is not valid.

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

CPF2469
Error occurred when sending message &1.

CPF247E
CCSID &1 is not valid.

CPF2488
Reply message queue *WRKSTN not valid for batch job.

CPF9830
Cannot assign library &1.

CPF9838
User profile storage limit exceeded.

SNDNETF (Send Network File) Command Description

SNDNETF Command syntax diagram

Purpose

The Send Network File (SNDNETF) command sends a save file or a member of a physical database file to another system user through the SNADS network. This command can be used to:

- Send data files to a user.
- Send source files to a user. Source sequence information is kept in the file sent.
- Send other object types stored in a save file to a user.

When the file arrives at its destination, a notification message is sent to both the recipient and sender of the file.

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

Restrictions:

1. The user must be enrolled in the system distribution directory.

2. The maximum size of a file that can be sent using the SNDNETF command is approximately 2 billion bytes.

Required Parameters

FILE Specifies the name of the file that is sent. The file being sent can be a physical file or a save file; logical files and device files are not allowed. Overrides to the specified file are ignored.

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

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

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

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

file-name: Specify the name of the file that is sent.

TOUSRID
Specifies the two-part user ID of one or more users to whom the file is being sent, or the name of one or more distribution lists containing the user IDs of one or more users to whom the file is being sent. A combination of user IDs and distribution lists can be specified on the same command. Each user ID or distribution list name is specified as a two-part name, and both parts are required.

Note: Depending on the type of work station being used, the internal value for a user identifier may differ from the characters shown by the Display Directory Entries (DSPDIRE) command. If the byte-string value specified for the TOUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

Optional Parameters

MBR Specifies the member that is sent from the file. A member name is not allowed if the file is a save file.

***FIRST:** The first member in the database file is used.

member-name: Specify the name of the file member that is sent.

TOTYPE
Specifies, when the user sends a source file, whether the sequence numbers and date fields are to be removed from the transmitted copy of the file. The source file is not changed. This parameter is not valid for nonsource files.

***FROMFILE:** The FILETYPE of the source file is used when sending the file. The transmitted file does not change.

***DATA:** The file is sent as a nonsource file. The transmitted copy is sent without sequence numbers and date fields.

CLASS

Specifies the VM/MVS SYSOUT class for distributions bound for a VM/MVS host system.

A: The class value is A.

class-value: Specify a class value. Valid values range from B through Z and 0 through 9.

FORMAT

Specifies the record format in which the network file is transmitted.

***V:** The file is sent using variable-length records with trailing blanks removed from each record.

***F:** The file is sent as fixed-length records with no trailing blanks removed from the records. Specifying this value affects only network files sent to a System/370 system. This value is not recommended for sending files to another iSeries 400.

***U:** If the file contains null capable fields, specifying this value causes SNDNETF to allow sending of the file and to send the mapping information which indicates which fields are null capable. This value for FORMAT will only produce the desired results when the file is sent to another AS/400 with OS/400 release VRM420 and later installed.

Note: Specifying FORMAT(*F) may increase the amount of storage and time required when transmitting the network file. Specifying FORMAT(*U) also requires that the receiving system must have a database file created with the identical field mappings as the file being sent, in order to receive the file correctly.

PTY Specifies the queuing priority used for this file when it is being routed through a SNADS network.

***NORMAL:** The file is sent with a service level priority of data low, which is used for most data traffic. On an iSeries 400, data low distributions are placed on the normal distribution queue specified for the route.

***HIGH:** The file is sent with a service level priority of data high, which is used for high priority data traffic. On an iSeries 400, data high distributions are placed on the data high distribution queue specified for the route.

Examples for SNDNETF

Example 1: Sending a Member

```
SNDNETF  TOUSRID((JONES SYSTEM1))  FILE(EMPLOYEE)
  MBR(PGMR)
```

This command sends member PGMR of file EMPLOYEE to the user identified to the network with a user ID of (JONES SYSTEM1). The library list is used to locate the file.

Example 2: Sending a Nonsource File

```
SNDNETF  TOUSRID((JONES SYSTEM2))  FILE(EMPLOYEE)
  MBR(PGMR) TOTYPE(*DATA)
```

This command sends member PGMR of file EMPLOYEE to the user identified to the network with a user ID of (JONES SYSTEM2). The library list is used to locate the file. The file is being sent as a nonsource file removing the sequence numbers and date fields.

Error messages for SNDNETF

*ESCAPE Messages

CPF8057

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

CPF8058
File &1 is a spooled file.

CPF8059
Member name not allowed for save file.

CPF8063
Cannot assign necessary resource.

CPF8064
File &1 in &2 member &3 not sent to any users.

CPF8066
One or more user identifiers on this command is not correct.

CPF8068
Error detected while processing file to be sent.

CPF8072
Object to be sent is greater than maximum size of 2GB.

CPF9005
System resource required to complete this request not available.

CPF9006
User not enrolled in system distribution directory.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

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

CPF9830
Cannot assign library &1.

CPF9845
Error occurred while opening file &1.

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

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

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

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

SNDNETMSG (Send Network Message) Command Description

SNDNETMSG Command syntax diagram

Purpose

The Send Network Message (SNDNETMSG) command sends a message to another system user through the SNADS network. This message is sent as an informational message to the message queue that is defined for the recipient on the receiving system.

Note: The recipient must have a valid message queue specified in his user profile. Messages sent with the SNDNETMSG command are rejected if the recipient does not have a message queue specified in the user profile. The message queue specified in the network attributes is not used. Additional information on specifying a message queue when sending and receiving messages is in the CL Programming book.

Restriction: The user must be enrolled in the system distribution directory. A description of the system distribution directory is in the SNA Distribution Services book.

Required Parameters

MSG Specifies the message text of the impromptu message that is sent. (An impromptu message is a message that is not stored in a message file.) Specify the message text. The text must be enclosed in apostrophes if it contains blanks or special characters. Up to 256 characters can be specified.

TOUSRID
Specifies one or more user IDs to whom an impromptu message is sent, or the name of one or more distribution lists containing user IDs of users to whom the message is to be sent. A combination of both user IDs and distribution lists can be specified on the same command. Each user ID or distribution list is specified as a two-part name, and both parts are required. Up to 50 user IDs can be specified.

Note: Depending on the type of work station being used, the internal value for a user identifier may differ from the characters shown by the Display Directory (DSPDIR) command. If the byte-string value specified for the TOUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

Example for SNDNETMSG

```
SNDNETMSG   MSG('Please do not make any more
  changes to the accounts receivable
  files for the next hour.')
  TOUSRID((SMITH SYSTEM2))
```

This command sends the message specified in the MSG parameter to the user identified to the network with a user ID of (SMITH SYSTEM2).

Error messages for SNDNETMSG

***ESCAPE Messages**

CPF8063
Cannot assign necessary resource.

CPF8066
One or more user identifiers on this command is not correct.

CPF8069
Message not sent to any users.

CPF9005
System resource required to complete this request not available.

CPF9006
User not enrolled in system distribution directory.

CPF9830
Cannot assign library &1.

CPF9845
Error occurred while opening file &1.

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

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

SNDNETSPLF (Send Network Spooled File) Command Description

SNDNETSPLF Command syntax diagram

Purpose

The Send Network Spooled File (SNDNETSPLF) command sends a spooled file to another system user on the Systems Network Architecture Distribution Services (SNADS) network. The file is placed on an output queue that is specified in the user profile of the user to whom the spooled file was sent.

When the file arrives at the destination system, a message is sent to both the recipient and sending user notifying them of the arrival of the spooled file.

Restrictions:

1. The user must be enrolled in the system distribution directory to run this command. The sender must have read, add, and delete authority to the receiving output queue when sending to user on the same system.
2. One of the following must be true:
   - The requester is the creator of the file.
   - The requester has *READ authority to the output queue on which the file resides, and DSPDTA(*YES) was specified on the CRTOUTQ command.
   - The requester has *SPLCTL special authority.
   - The requester has *JOBCTL special authority, and the output queue on which the file resides has OPRCTL(*YES) specified on the CRTOUTQ command.
   - The output queue has DSPDTA(*YES) specified on the CRTOUTQ command.
   - The requester has owner authority to the output queue on which the file resides and the queue had AUTCHK(*OWNER) and DSPDTA(*YES) or DSPDTA(*NO) specified on the CRTOUTQ command.

- The requester has *READ, *ADD, and *DELETE authority to the output queue on which the file resides and the queue has AUTCHK(*DTAAUT) and DSPDTA(*YES) or DSPDTA(*NO) specified on the CRTOUTQ command.

3. DTAFMT(*RCDDATA) must be used when sending a spooled file to a release prior to Version 1 Release 3 Modification 0 (V1R3).

Required Parameters

FILE Specifies the name of the spooled file that is sent to the specified user. The file name is the name of the device file that was used by the program to produce the spooled file. Specify the name of the spooled file.

TOUSRID
Specifies the user ID of one or more users to whom the spooled file is sent, or the name of one or more distribution lists containing the user IDs of one or more users to whom the spooled file is sent. A combination of user IDs and distribution lists may be specified on the same command. Each user ID or distribution list name is specified as a two-part user name, and both parts are required. The users in the distribution list may be either remote or local.

Note: Depending on the type of work station being used, the internal value for a user identifier may differ from the characters shown by the DSPDIR command. If the byte-string value specified for the TOUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

Optional Parameters

JOB Specifies the name of the job that created the spooled file whose data records are sent. If no job qualifier is given, all jobs currently in the system are searched for the simple job name.

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 command is the job that created the spooled file.

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 spooled file.

job-number: Specify the number of the job that created the spooled file.

SPLNBR
Specifies the number of the spooled file from the job whose data records are to be copied. 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 having the specified file name whose data records are copied.

» **JOBSYSNAME**

Specifies the name of 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, spooled file number, and creation date and time is used.

***ANY:** The job system name is not used to determine which spooled file is used. Use this value when the spooled file 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 the job which created the spooled file ran on.

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, 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. «

DTAFMT

Specifies the format in which to send a spooled printer file. This parameter is valid only for spooled printer files; it is not valid for diskette files.

If DTAFMT(*RCDDATA) is specified, then a spooled file that contains special device requirements cannot be sent.

Note: The device requirements are listed as part of the attributes for the spooled file. To view the device requirements, use the WRKSPLFA command or the attributes option on the WRKSPLF command. If any of the device requirements are attributes of the file (if any of the device requirements on the display have a 'Y'), the spooled file cannot be sent. Either specify DTAFMT(*ALLDATA) or copy the spooled file to a database file using the Copy Spooled File (CPYSPLF) command and then use the Send Network File (SNDNETF) command to send the file as a data file.

If DTAFMT(*RCDDATA) is specified, the following attributes of the spooled file are kept:

- File name
- Number of copies
- Characters per inch
- Drawer
- Form type
- Lines per inch
- Print text
- Page length
- Page rotation
- Page width
- Font name
- Diskette label
- Diskette creation and end dates
- Diskette code type
- Diskette exchange type

***RCDDATA:** The spooled printer file is converted to a format acceptable for releases prior to V1R3 of the AS/400 system, including System/36, System/38, and System/370 systems. A limited set of attributes are sent.

***ALLDATA:** The spooled printer file is sent in the same format as it is stored on the spool. This value is only compatible with V1R3 or later releases of the AS/400 system.

Notes:

1. If a file will be printed on the receiving system, it must be printed on the same type of printer as it was intended to be printed on the source system.
2. Spool files that require advanced function printing (AFP) resources may print differently on the receiving system. To assure that spool files are printed in the same manner, the following must occur:
   - Before using this command to send a file to a user on the same system, make sure that the libraries containing non-IBM supplied AFP resources are in the library list.
   - Before using this command to send a file to a different user or system, make sure that the libraries containing non-IBM supplied AFP resources are in the initial library list of the user receiving the spooled files.
3. Use this format to send *LINE, *AFPDS, and *AFPDSLINE printer type device files to System 370.

CLASS

Specified the VM/MVS SYSOUT class for distributions bound for a VM/MVS host system.

A: The class is A.

class-value: Specify a distribution class value. Valid values range from B through Z and 0 through 9.

PTY Specifies the queuing priority used for this spooled file when it is being routed through a SNADS network.

***NORMAL:** The spooled file is sent with a service level priority of data low, which is used for most data traffic. On an iSeries 400, data low distributions are placed on the normal distribution queue specified for the route.

***HIGH:** The spooled file is sent with a service level priority of data high, which is used for high priority data traffic. On an iSeries 400, data high distributions are placed on the data high distribution queue specified for the route.

Examples for SNDNETSPLF

Example 1: Sending a Spooled File

```
SNDNETSPLF  FILE(QPRINT)  TOUSRID((JDE SYS1))
  JOB(142857/PAPER/PRINT)  SPLNBR(*LAST)
  DTAFMT(*ALLDATA)
```

This command sends the last (most recently created) copy of spooled file QPRINT from job 142857/PAPER/PRINT to the user with a user ID of JDE SYS1. All spooled file functions will be sent.

Example 2: Sending Print Attributes

```
SNDNETSPLF  DTAFMT(*RCDDATA)
```

This command sends a limited set of print attributes.

Example 3: Sending All Print Attributes

```
SNDNETSPLF  DTAFMT(*ALLDATA)
```

This command sends all print attributes. *ALLDATA is only valid when it is sent from one iSeries 400 to another iSeries 400, and both systems are Release 3.0 or higher.

Error messages for SNDNETSPLF

*ESCAPE Messages

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

CPF3205
File not created.

CPF3207
Member not added. Errors occurred.

CPF3303
File &1 not found in job &5/&4/&3.

CPF3309
No files named &1 are active.

CPF3330
Necessary resource not available.

CPF3340
More than one file with specified name found in job &5/&4/&3.

CPF3342
Job &5/&4/&3 not found.

CPF3343
Duplicate job names found.

CPF3344
File &1 number &2 no longer in the system.

CPF3429
File &1 number &2 cannot be displayed, copied, or sent.

CPF3482
Copy request failed. Spool file &1 is open.

CPF3486
CHLVAL parameter value not valid.

CPF3492
Not authorized to spooled file.

CPF3493
CTLCHAR parameter not correct for file &1.

CPF3499
Records in file &1 preceded all assigned channel values.

CPF8055
Spooled file contains special device requirements. File not sent.

CPF8063
Cannot assign necessary resource.

CPF8066
One or more user identifiers on this command is not correct.

CPF8067
File &1 not sent to any users.

CPF8068
Error detected while processing file to be sent.

CPF8072
Object to be sent is greater than maximum size of 2GB.

CPF9005
System resource required to complete this request not available.

CPF9006
User not enrolled in system distribution directory.

CPF9820
Not authorized to use library &1.

CPF9830
Cannot assign library &1.

CPF9845
Error occurred while opening file &1.

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

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

CPF9849

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

»

SNDPRD (Send Product) Command Description

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

SNDPRD Command syntax diagram

Purpose

The Send Product (SNDPRD) command provides the capability to send a product, packaged with the OS/400 packaging support, from the central site system to one or more managed systems.

Note: A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

Restrictions:

1. This command is shipped with public authority of *EXCLUDE.
2. Because this command calls the PKGPRDDST and ADDPRDCRQA commands, you must be authorized to use these commands. You must have *ALL authority to folders.
3. If a NODL value is specified, the node list can contain only entries that have a value of *SNA for the address type.
4. The Managed System Services licensed program and the OS/400 licensed program cannot be specified.

Notes

1. This command packages the product for distribution if it is not already packaged.
2. If the product was packaged for distribution before but the PKGPRDDST(*REPLACE) parameter was specified, this command packages the product for distribution again. It replaces the previous packaged product.
3. When it is requested to send the license key of the product, a record with the license key information must exist in the central site license repository by the time this activity runs.
4. All the existing records in the central site license repository containing the license key information for the specified product will be sent to the specified managed system or systems, but only those that match the system serial number are added to the managed system license repository.

Required Parameter

PRDID

Specifies the 7-character identifier of the product to be sent.

product-ID: Specify the 7-character product ID that is used in the activity.

Optional Parameters

RLS Specifies which version, release, and modification level of the product is used.

***ONLY:** Only one version, release, and modification level is used for the product option.

release-level: Specify the release level in the format VxRxMy, where Vx is the version number, where Rx is the release number, and My is the modification number. Valid values for *x* range from 0 through 9. Valid values for *y* range from 0 through 9 and the letters A through Z.

OPTION

Specifies which of the optional parts of the product given in the PRDID parameter are used.

***BASE:** Only the base part of the product is used.

product-option-number: Specify the option number for the product load being saved. Valid values are from 1 through 99.

LODTYPE

Specifies the product load objects being used.

***ALL:** Code and language objects specified on the LODID parameter are used.

***CODE:** The program objects associated with this product load are used.

***LNG:** The objects associated with the NLV identified on the LODID parameter are used.

LODID

Specifies the load identifier used.

***ALL:** All languages for this product option are saved.

***CODE:** The code load is to be used.

product-load-ID: Specify the load ID of the product when LODTYPE(*LNG) or LODTYPE(*ALL) is used. The load ID must be one of the valid IBM national language versions and be specified in the form 29xx. The value for x can be from 0 through 9.

TGTRLS

Specifies the release of the operating system on which you intend to use the product.

***CURRENT:** The product is to be used on the release of the operating system currently running on your system.

***PRV:** The product is to be used on the previous release.

release-level: Specify the release level in the format VxRxMx. The product can be used on a system with the specified release or with any later release of the operating system installed.

Valid values depend on the current version, release, and modification level, and can change with each new release.

NODL Specify the node list object name that contains a list of systems that are the destinations for the request. Node list name cannot be specified if a value is specified for the control point name (CPNAME) parameter.

***NONE:** The systems on which this activity is to be performed are not identified by a node list. Individual control point names must be specified.

The possible library values are:

***LIBL:** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

***CURLIB:** The current library for the job is used to locate the node list object.

library-name: Specify that only the library named in this parameter is searched.

node-list-name: Specify the node list object name that contains the list of systems on which the command is to be performed.

CPNAME

Specifies the APPN control point names of the managed systems on which this request is to be performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

***NONE:** The systems on which this activity is to be performed are not identified individually. A node list must be specified.

***NETATR:** The network ID of the local system is used. This is useful when the node being specified is in the same network as the local system.

network-identifier: Specify the APPN network identifier of the managed system on which the activity is to be performed. It is an 8-character field.

control-point-name: Specify the APPN control point name of the managed system on which the activity is to be performed.

INSTALL

Specifies that the product is installed on the managed system when it arrives.

***NO:** The product is not installed.

***YES:** The product is installed.

KEEPCLGE

Specifies if the catalog entry and associated save file corresponding to the product is kept on the specified system or systems. This is only valid if INSTALL(*YES) was specified.

***NO:** The catalog entry and associated save file are not kept.

***YES:** The catalog entry and associated save file are kept.

SNDLICKEY

Specifies if the license key is to be sent with the product.

***YES:** The license key is sent with the product.

***NO:** The license key is not sent with the product.

PKGPRDDST

Specifies that the product is packaged for distribution before it is sent. This parameter specifies whether to replace the packaged product if it already exists in the distribution repository.

***NOREPLACE:** Specify that the existing packaged product is not replaced.

***REPLACE:** Specify that the existing packaged product is replaced.

Examples for SNDPRD

Example 1: Sending a Product

```
SNDPRD  PRDID(1ACCOUN)   RLS(V5R2M0)   OPTION(*BASE)
  LODTYPE(*ALL)  LODID(*ALL)  CPNAME((*NETATR SYS1))
  INSTALL(*YES)  SNDLICKEY(*YES)
```

Send 1ACCOUN product with base option release V5R2M0 for both the program and the language parts, and install it on the iSeries server SYS1. The license key for this product is also sent and is added to the managed system license repository.

Example 2: Sending but Not Installing an Option

```
SNDPRD  PRDID(1CHECKS)  RLS(V5R2M0)   OPTION(0010)
  LODTYPE(*ALL)  LODID(*ALL)  CPNAME((*NETATR SYS2))
  SNDLICKEY(*YES)
```

For both the program and the language parts, send but do not install the option 10 of the 1CHECKS product, release V5R2M0, on the iSeries server SYS2. The license key for this product is also sent and is added to the managed system license repository.

Error messages for SNDPRD

***ESCAPE Messages**

None «

SNDPGMMSG (Send Program Message) Command Description

SNDPGMMSG Command syntax diagram

Purpose

The Send Program Message (SNDPGMMSG) command sends a message to a named message queue or to a call message queue. A call message queue can be the *EXT external message queue or a message queue associated with a call stack entry. Each time a program or procedure is called a new message queue is associated with its call stack entry. The message queue is identified by the name of its associated program or procedure.

A program can send a message to its own message queue or to a message queue that is associated with a different call stack entry.

This command can send both exception and non-exception messages. When an exception message is sent, processing of the sending program is interrupted until action is taken on the exception. Depending on the action taken, processing of the sending program may resume. If allowed to resume, processing continues in the sending program at the CL command immediately following the SNDPGMMSG command. When a non-exception message is sent, processing of the sending program is not interrupted. After the message is sent, the sending program continues to run at the CL command immediately following the SNDPGMMSG command.

When a program or procedure monitors for an exception message and an exception message is sent to the call stack entry in which that program or procedure is running, the related exception handler is called and run by the system. Therefore, a program or procedure can identify when an exception message is sent to the call stack entry in which it is running. There is no automatic notification for non-exception messages.

The message can be an immediate or a predefined message that is stored in a message file. Substitution data can be specified in the MSGDTA parameter (as a single character string containing one or more concatenated message data fields) to replace the substitution variables in the message. The message can be sent to the job's *EXT message queue, to a message queue associated with a call stack entry, to one or more nonprogram message queues (such as a user profile message queue), to the system operator message queue, or to the system history log, QHST.

Restrictions:

1. This command is valid only in CL programs.
2. The SNDPGMMSG command allows a message of up to 512 characters to be sent. However, if the message is sent to the *EXT message queue of an interactive job, only 76 characters are shown on

the Display Program Messages display. If the message is sent to a user's, work station's, or the system operator's message queue, the Display Message (DSPMSG) command allows all 512 characters to be displayed.

3. This command can only send inquiry messages (specified by MSGTYPE(*INQ)) to one message queue or to two nonprogram message queues if one of the queues is *HSTLOG.

Required Parameters

MSG Specifies the message text that is sent. A maximum of 3000 characters can be specified or, if you are prompting for this command, a maximum of 512 characters can be specified. The string must be enclosed in apostrophes if special characters (including blanks) are used. If a value is specified on this parameter, the MSGID, MSGF, and MSGDTA parameters cannot be specified.

Coded Character Set Identifier (CCSID) Considerations

The text supplied on the MSG parameter is assumed to be in the CCSID of the job running this command unless a coded character set identifier is supplied in the CCSID parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center.

MSGID

Specifies the message identifier of a message description whose predefined message is being sent by the program to a message queue. If MSGID is specified, the MSG parameter cannot be specified.

MSGF Specifies the qualified name of the message file that contains the predefined message being sent. This parameter is required if MSGID is specified.

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

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

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

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

message-file-name: Specify the name of the message file that contains the predefined message.

Optional Parameters

MSGDTA

Specifies the character string, or a CL variable that contains a character string, containing one or more substitution values that are used as message data fields within the predefined message. The substitution values take the place of the substitution variables that were defined in the message text when the message was defined.

Note: MSGDTA is not valid if MSG is specified.

Rules for Replacing CL Variables with Character Strings

- Multiple values must be concatenated to form a single character string (more information is in the CL Programming book).

- If more than one substitution variable exists in the message, the specified character string must be the same length as the sum of the message data fields defined in the message.
- The length of the entire character string of concatenated substitution values cannot be greater than 3000 characters, or if you are prompting for this command, the length cannot be greater than 512 characters. More information about the length of individual message data fields is in the FMT parameter description in the ADDMSGD (Add Message Description) command.
- Each substitution value, specified in the same order in the string as defined in the FMT parameter of the ADDMSGD command, must be specified but only if its associated variable was defined.
- If the length of the message data that is passed to a substitution variable is shorter than the length specified for the FMT parameter of the ADDMSGD command, the substitution value becomes a null field.
- If the length of the message data passed to a substitution variable is longer than the length specified for ADDMSGD FMT, the message data is truncated.

Note: If the last character in the string passed to the substitution variable is blank (for the format types *BIN, *ITV, or *DTS, a blank is indicated by the hex value 40), the system truncates the blank. If the resulting string is shorter than the length specified on the FMT parameter of the ADDMSGD command, the substitution variable becomes a null field.

To prevent the system from truncating the blanks, concatenate a nonblank character (for the format type *BIN, *ITV, and *DTS, add value other than 40) to the string after the blanks. The system retains the blanks and truncates the additional character.

***NONE:** No message data is supplied for the message.

character-string: Specify either the character string that gives the substitution values in the specified predefined message sent by the program, or the name of the CL variable that contains the character string.

If the message contains more than one substitution variable, the character string containing the replacement values must be coded with the proper fill characters (zeros for numeric values, and blanks for character values) to position the values properly in the string to match the concatenated fields used to create the single string. If the string contains special characters (including blanks), it must be enclosed in apostrophes. The null value *N cannot be specified for a substitution value.

For example, if a predefined message contains the following three substitution variables:

Variable	Format Type	Length	Expected Values
&1	*CHAR	3 characters	Yes or no
&2	*CHAR	4 digits	4 digits
&3	*CHAR	8 characters	One or two words

the following valid character strings can be specified:

```
MSGDTA('Yes0007not done')
MSGDTA('No 0285finished')
MSGDTA(&DTA)
```

Note that the variable &2 is defined as a character variable, not as a decimal variable. If a character string is to be used instead of a CL variable to specify the substitution values in the

MSGDTA parameter, the associated substitution variables must be defined (in the FMT parameter of the ADDMSGD command) as character variables to prevent results that are not valid in the text of the message that is sent.

The last example shows, instead of the actual character string, the CL variable &DTA, whose value is used as the message data. &DTA might have been declared in the sending program as:

```
DCL   VAR(&DTA)  TYPE(*CHAR)
  LEN(15)  VALUE('YES0175not done')
```

The character string, specified as the initial value for &DTA, contains the three concatenated substitution values `YES`, `0175`, and `not done`, which are to be sent as message data.

Coded Character Set Identifier (CCSID) Considerations

The text supplied on the MSGDTA parameter that corresponds to the *CCHAR type field is assumed to be in the CCSID of the job running this command unless the coded character set identifier is supplied in the CCSID parameter. All other text supplied on the MSGDTA parameter is assumed to be 65535 and is not converted. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center. For more information about the *CCHAR type fields, see the Add Message Description (ADDMSGD) command.

TOUSR

Specifies that the message is to be sent to the message queue specified in the user profile for the user named on this parameter. This parameter cannot be used if TOMSGQ or TOPGMQ is specified.

***SYSOPR:** The message is sent to the system operator's message queue, QSYS/QSYSOPR. Any message sent to QSYSOPR automatically has a copy of the message sent to QHST.

***REQUESTER:** The message is sent to the user profile's message queue for interactive jobs or to the system operator's message queue (QSYS/QSYSOPR) for batch jobs.

***ALLACT:** A copy of the message is sent to the user profile message queue of each user profile with an interactive job currently running. This value cannot be specified if an inquiry message is being sent (the value *INQ is specified in the MSGTYPE parameter).

user-profile-name: Specify the name of the user profile of the user to whom the message is sent.

TOMSGQ

Specifies the qualified names of up to 50 nonprogram message queues to which an informational message is sent. For an inquiry message, one message queue may be specified or two message queues may be specified if one of the queues is *HSTLOG. This parameter cannot be used if TOUSR is specified.

***TOPGMQ:** The message is sent only to the call message queue specified on the TOPGMQ parameter.

***SYSOPR:** The message is sent to the system operator's message queue, QSYS/QSYSOPR. Any message sent to QSYSOPR automatically has a copy of the message sent to QHST.

The name of the message 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.

message-queue-name: Specify the names of the message queues to which the message is sent. Up to 50 names can be specified.

***HSTLOG:** The message is sent to the system history log message queue, QSYS/QHST. If *HSTLOG is specified more than once, only one message will be sent to QSYS/QHST. If *HSTLOG is specified with QSYSOPR only one message is sent to QSYS/QHST.

TOPGMQ

Specifies the call message queue to which the message is sent. The call message queue can be the *EXT external queue or the call message queue associated with a call stack entry.

Element 1: Relationship

Element 1 of this parameter specifies whether the message queue is associated with the call stack entry identified by Element 2, or if it is associated with the call stack entry that is immediately previous to the one identified by Element 2.

***PRV:** The message is sent to the message queue of the call stack entry that is immediately previous to the one identified by Element 2. However, if the message queue immediately previous to the one identified by Element 2 is for an Integrated Language Environment (ILE) program entry procedure (PEP), the message is sent to the message queue that precedes the PEP message queue in the stack.

***SAME:** The message is sent to the message queue of the call stack entry identified by Element 2.

Element 2: Program or Qualified Procedure

Element 2 is used to provide the information needed to identify a call stack entry. A call stack entry can be identified by specifying the name of the OPM program or ILE procedure that is running in that entry. It can also be identified by using a special value or a special value used in conjunction with a name.

Element 2 of this parameter has three items. Item 1 specifies an OPM program or ILE procedure name or a special value. Item 2 specifies an ILE module name which is used as a qualifier for the value specified in item 1. Item 3 can specify either an OPM program name or an ILE program name depending on what is specified in item 1. Item 3 is also used as a qualifier for what is specified in item 1.

Item 1: Program or Procedure Name

***:** Identifies the OPM program or ILE procedure running the SNDPGMMSG command.

OPM-program or ILE-procedure name: Specify the OPM program name or ILE procedure name which will be used to identify the call stack entry. If item 1 is to identify an OPM program, the name specified can be a maximum of 10 characters. If item 1 is to identify an ILE procedure, the name specified can be a maximum of 256 characters. The system begins its search for the specified program or procedure with the most recently called program or procedure.

If an ILE procedure name is specified, that name may be further qualified by using items 2 and 3 to specify an ILE module name and an ILE program name. The module name identifies the module that procedure was compiled into. The program name identifies the program that the procedure was bound into. Either one or both of these items may be specified. When a module name or program name is specified, the corresponding call stack entry will not only be running a procedure by the specified name but that procedure will also be part of the specified module and specified program. 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.

***CTLBDY:** Specifies the call stack entry that is at the most recent control boundary. This entry will be running in the same activation group as the CL program that is running the SNDPGMMSG command. Note that a control boundary will not exist if all programs on the call stack are OPM programs.

***PGMBDY:** Specifies the program boundary of either the program that is using the SNDPGMMSG command or the program whose name is specified in item 3. If no name is specified in item 3, it is assumed that the program is the one using the command.

If it is an ILE program that is being specified, this special value identifies the call stack entry for the program entry procedure (PEP) of that program, if the program was called by a dynamic call. If the program was called by a procedure pointer, this special value identifies the call stack entry for the procedure that was pointed to.

If it is an ILE service program that is being specified, this special value identifies the call stack entry for the first procedure that was called in that service program.

If the program being specified is an OPM program, this special value has the same effect as specifying the special value * or a program name for item 1. A difference will occur if the OPM program has called itself recursively. In this case, this special value identifies the first recursion level rather than the current recursion level as would be the case if the special value * or a program name was used.

***PGMNAME:** Specifies that the call stack entry will be identified only by using a program name and optionally a module name. When this special value is used, item 3 must specify an ILE program or service program name or OPM program name. Item 2 may contain either the special value *NONE or an ILE module name.

This special value is used to send a message to the most recently called procedure that is part of the specified ILE program or service program. When using this special value, it is not necessary to explicitly provide a procedure name. If a module name is also provided, then this special value is used to send a message to the most recently called procedure that is both part of the identified program and the identified module.

This special value may also be used to send a message to an OPM program. In this case, using this special value and providing the OPM program name in item 3 has exactly the same effect as providing that program name here in item 1. Note that if this special value is being used to send to an OPM program then the module name must be specified as *NONE.

Item 2: ILE Module Name

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

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

ILE-module-name: Specify the ILE module name that is to be used as a qualifier for the value specified in item 1. A module name can be specified if item 1 contains an ILE procedure name or the special value *PGMNAME. The module name can be a maximum of 10 characters.

Use the special value *NONE when a module name is not being specified but a program name is in item 3.

Item 3: OPM Program Name or ILE Program Name

The ILE program or service program name or OPM program name that is to be used in conjunction with the procedure name or special value that is specified in item 1.

***NONE:** No ILE or OPM program name is specified.

ILE-program-name: Specify the ILE program or service program name that is to be used as a qualifier for the value specified in item 1. An ILE program or service program name can be specified if item 1 contains an ILE procedure name or one of the special values of *PGMBDY or *PGMNAME. The ILE program or service program name can be a maximum of 10 characters.

OPM-program-name: Specify the OPM program name that is to be used as a qualifier for the value specified in item 1. An OPM program name can be specified if item 1 contains the special value of *PGMBDY or *PGMNAME. The OPM program name can be a maximum of 10 characters.

Single Values

***EXT:** The message is sent to the external message queue of the job. The external message queue is used to communicate with the external requester of the job, such as a display station user. *INQ messages that are sent to *EXT in an interactive job wait for 24 hours before the default reply is sent. In a batch job the default reply is sent immediately.

Messages sent to this queue can be 512 characters in length, but only 76 characters of text are shown on the Program Messages display.

MSGTYPE

Specifies the message type assigned to a message sent by this program.

Notes:

1. Inquiry messages can be sent only to the external queue or to a named message queue specified on the TOUSR or TOMSGQ parameters. When sending an inquiry with the TOMSGQ parameter, a second queue can be specified if the value is *HSTLOG.
2. Completion, diagnostic, escape, notify, and status messages can be sent only to a call message queue.
3. Escape messages cannot be sent to the external message queue.

***INFO:** The message is sent as an informational message.

***INQ:** The message is sent as an inquiry message.

***COMP:** A completion message is sent to a call message queue. A completion message indicates the status of the work that is successfully performed.

***DIAG:** A diagnostic message is sent to a call message queue. Diagnostic messages provide information about errors detected by this program. The errors are either in the input sent to it, or are those that occurred while it was running the requested function. An escape or notify message should also be sent to inform the receiving program or procedure of the diagnostic messages that are on its message queue.

***NOTIFY:** A notify exception message is sent to a call message queue. A notify message describes a condition for which corrective action must be taken before the sending program can continue. A reply message is sent back to the sending program. After corrective action is taken, the sending program can resume running and can receive the reply message from its message queue.

***ESCAPE:** An escape exception message is sent to a call message queue. An escape message describes an irrecoverable error condition. The sending program does not continue to run.

***RQS:** A request message is sent to a call message queue. A request message allows request data received from device files to pass from this program to another program or procedure. An immediate message, specified by the MSG parameter, must be used to send the request.

***STATUS:** A status exception message is sent to a call message queue. The status message describes the status of work performed by the sending program. The first 28 characters of message data in the MSGDTA parameter are used as the comparison data for message monitors (established by the Monitor Message (MONMSG) command). If the status exception message is not being monitored, control is returned to the sending program. If a status message is sent to the external message queue of an interactive job, the message is shown on line 24, processing continues, and no response is required.

Note: This value cannot be specified if the MSG parameter is specified.

RPYMSGQ

Specifies, for inquiry and notify messages only, the name of the call message queue or the qualified name of the nonprogram message queue to which a reply message is sent. If a notify message is sent, or an inquiry message is sent to *EXT, the reply message queue must specify *PGMQ.

***PGMQ:** The reply to an inquiry or notify message is sent to the message queue associated with the call stack entry of the program or procedure using this command.

The name of the nonprogram message 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.

message-queue-name: Specify the name of the message queue, that is not for a program, to which the reply is sent.

CCSID

Specifies the coded character set identifier (CCSID) that the supplied message or message data is in. If a message identifier is specified, the text supplied by the MSGDTA (message data) parameter that corresponds to the *CCHAR type field is assumed to be in the CCSID supplied by the CCSID parameter. The data supplied that does not correspond to the *CCHAR type field is assumed to be 65535 and is not converted. For more information about the *CCHAR type field see the Add Message Description (ADDMSGD) command.

If no message identifier is specified, the text specified by the MSG (message) parameter is assumed to be in the CCSID specified by the CCSID parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center.

***JOB:** The message data or immediate text is assumed to be in the CCSID of the job running this command.

***HEX:** The message data or immediate text is not converted. CCSID 65535 is used.

coded-character-set-identifier: Specify a valid CCSID in which you want your message or message data to be considered in. Valid values range from 1 through 65535. This command validates the CCSID. See the Globalization topic in the Information Center for a list of valid CCSID values.

KEYVAR

Specifies the name of the CL character variable, if any, that contains the message reference key that identifies the message sent by the program containing this SNDPGMMSG command. The message reference key is assigned by the system when the message is sent and is placed in the variable specified by KEYVAR. The message key can then, for example, be used in the MSGKEY parameter of the RCVMSG command to receive the reply to an inquiry message being sent by this command. The message reference key can be used by the program specified on the TOPGMQ parameter 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.

If a message is being sent to a message queue associated with a call stack entry, KEYVAR refers to that message queue (specified by the TOPGMQ parameter). If *INQ or *NOTIFY is specified for the MSGTYPE parameter, KEYVAR refers to the message queue specified in the RPYMSGQ parameter. In all other cases, KEYVAR refers to the message queue specified in the TOMSGQ parameter.

If an inquiry message has been sent to the *EXT message queue with a value specified for the KEYVAR parameter, the message reference key returned is that of the sender's copy message, which is located on the reply message queue. The reply message queue defaults to the program message queue of the program that sent the inquiry message.

Any type of message can be assigned a key when it is being sent to a program message queue. For messages sent to a nonprogram message queue, message reference keys are available for inquiry (*INQ) messages only. If another message type is sent to a nonprogram queue, no message key is available and blanks are returned for KEYVAR.

The variable must be a character variable having a length of 4 characters. If KEYVAR is not specified and a reply is required, it can be received by the program in FIFO order.

Examples for SNDPGMMSG

Example 1: Specifying Substitution Values

```
SNDPGMMSG   MSGID(UIN0023)  MSGF(INV)
  MSGDTA('50 100')  TOPGMQ(*EXT)
```

This command sends the message identified as UIN0023, which is stored in message file INV, to the external message queue of the job (the Display Program Messages presents the message at a display station). The data, which contains two substitution values specified in the MSGDTA parameter, is sent with the message. This data can then be used as substitution values when the message is received, or it can be used as data to be dumped, depending on how the message UIN0023 is defined in the message file. Assuming that the variables &1 and &2 have been defined in the message file as character variables, each 3 characters long, and that the first-level message text of the message UIN0023 is: 'Requested item decreased by &1; current balance &2.' The message text sent is: 'Requested item decreased by 50; current balance 100.'

Example 2: Sending an Inquiry Message

```
SNDPGMMSG   MSG('Mount payroll checks in printer
  before continuing')  MSGTYPE(*INQ)
  TOMSGQ(*SYSOPR)
```

This command sends an inquiry message to the system operator. The operator looks at the message that was sent by using the DSPMSG command and responds to the message directly on that display. A Receive Message (RCVMSG) command is used in the program to accept the operator's response.

Example 3: Sending an Escape Message

```
SNDPGMMSG   MSGID(USR0001)  MSGF(USRMSGR)
  TOPGMQ(*PRV  *)  MSGTYPE(*ESCAPE)
```

This command is an example of how a message could be sent to the caller of a program or procedure to cause an abnormal end. The message USR0001 could indicate that an invalid code was passed (such as a nonnumeric code when numeric is required). Because the message being sent is an escape message, the program or procedure that is sending the message cannot be resumed. The values *PRV and * did not have to be coded on this command because they are the default values on the TOPGMQ parameter.

Example 4: Sending an Escape Message to an ILE Procedure

```
SNDPGMMSG   MSGID(USR0001)  MSGF(USRMSGR)
  TOPGMQ(*SAME ACCOUNT_FINAL_TOTALS)  MSGTYPE(*ESCAPE)
```

This command sends a message to an ILE procedure. In this example, the call stack entry identifier is more than 10 characters. Since no qualifier is specified, the actual module name and bound program name associated with the procedure are not used in finding the procedure. The escape exception message is sent to the message queue associated with ACCOUNT_FINAL_TOTALS because *SAME is specified for Element 1.

Example 5: Sending an Escape Message using Qualifiers

```
SNDPGMMSG   MSGID(USR0001)  MSGF(USRMSGR)
  TOPGMQ(*PRV FIRST_QTR_SUMMARY SUMQTRS REPORTS)
  MSGTYPE(*ESCAPE)
```

This command sends an escape exception message to the caller of the procedure FIRST_QTR_SYMMARY. The procedure is qualified by the module name SUMQTRS and the bound program name REPORTS. The escape exception message interrupts the sending program and the sending program is not resumed.

Example 6: Sending a Completion Message using a Partial Procedure Name

```
SNDPGMMSG   MSGID(USR0001)  MSGF(USRMSGR)
  TOPGMQ(*SAME MANAGE_SALES>>>)
  MSGTYPE(*COMP)
```

This command sends a completion message to the most recent procedure whose name begins with MANAGE_SALES.

Error messages for SNDPGMMSG

*ESCAPE Messages

CPF24CB
*PGMNAME requires a specified program name.

CPF2409
Specified message type not valid with specified program message queue.

CPF2428
Message queue parameter is not valid.

CPF2453
Reply queue not sender’s program message queue.

CPF2469
Error occurred when sending message &1.

CPF247A
Call stack entry not found.

CPF247E
CCSID &1 is not valid.

CPF2499
Message identifier &1 not allowed.

CPF2524
Exception handler not available because of reason code &1.

CPF2550
Exception message sent to a deleted program or procedure.

CPF2702
Device description &1 not found.

CPF7C08
No support network connection.

CPF8C0C
Content of problem record &1 not valid.

CPF8C0E
Library QGPL not found.

CPF8C01
Cannot connect to IBM service system. One session allowed.

CPF8C07
A parameter is not valid.

CPF8C08
Cannot specify *SELECT for the control point name.

CPF8C09
&1 not defined as a service provider.

CPF8C16
Error occurred while processing request.

CPF8C17
Sign-on failed.

CPF8C18
No support network connection.

CPF8C19
Remote support application failed.

CPF8C2A
Cannot connect to IBM service system.

CPF8C24
Error occurred while processing request.

CPF8C27
Alternate load device not found.

CPF8C32
PTF order cannot be processed.

CPF9830
Cannot assign library &1.

CPF9845
Error occurred while opening file &1.

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

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

»

SNDPTF (Send Program Temporary Fix) Command Description

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

SNDPTF Command syntax diagram

Purpose

The Send Program Temporary Fix (SNDPTF) command sends a specific program temporary fix (PTF) or a list of PTFs to a specific service requester or a list of service requesters. Cover letters can also be sent if they are available.

When using the SNDPTF command, you can request that the PTFs be applied on the service requester when the PTFs arrive on the service requester. When requesting that the PTFs be applied, you can also request an IPL of the service requester's system. The apply PTF and IPL can be scheduled to occur at a later date.

Notes:

1. A change request is automatically submitted that can be viewed to determine the status of this command. You can use the Work with Submitted Change Request (WRKSBMCRQ) command to monitor the status.
2. A maximum of 300 PTFs can be specified.

Restrictions:

1. The apply and IPL functions of this command are only supported when the service requester has Managed System Services licensed program installed.
2. A PTF save file must exist and be released in order to be sent another system.
3. PTFs that are not marked as delayed are applied immediately. PTFs marked as delayed are scheduled to be applied at the next IPL.
4. PTFs marked as delayed can only be applied permanently if they have been previously applied temporarily. Notice that an IPL would have been required to apply them because they are delayed.
5. If a value for NODL is specified, the node list can contain only entries that have a value of *SNA for the address type.
6. The product and release of the PTF must be supported using the Work with Supported Products (WRKSPTPRD) command.
7. If you are distributing a PTF and the distribution queue is set to *SNADS in the Work with Service Requesters (WRKSRVRQS) command, you must have *USE authority to the Send PTF (SNDPTF) command.
8. If you are distributing a PTF and the distribution queue is set to *SVDS in the Work with Service Requesters (WRKSRVRQS) command, you must have *USE authority to the Copy PTF (CPYPTF) command.

Required Parameters

PTFID Specifies the PTF numbers for the PTFs that are to be sent. The following special PTFs can also be distributed:

SF97vrm or nnn
PTF summary package

MF98vrm or nnn
Hardware service planning tips

SF98vrm or nnn

Software service planning tips

```
n is the PTF sequence number
v is the version for the package requested
r is the release for the package requested
m is the modification level
  for the package requested
```

Element 1: PTF Identifier

ptf-identifier: Specify the 7-character identifier (ID) of the PTF that is sent.

Element 2: Product Identifier

SPT: The PTF identifier is unique to only one supported product.

product-id: Specify the product ID of the PTF that is to be sent. The product must be specified when PTF identifiers are not unique across products.

Element 3: Release Level of Product

ONLY: The PTF identifier is unique to only one release level of the product.

release (VxRxMx): Specify the version, release, and modification level of the PTF being sent. The release level must be specified when the PTF identifiers are not unique across products. The format is VxRxMx.

DESTSRVRQS

Specifies the service requester system where this PTF function is performed. A single service requester or a list of service requesters can be specified.

Element 1: Network Identifier

SELECT: Shows a list of service requesters. From the list, you can select one or more service requesters. *SELECT is not valid when the command is used in a batch environment.

ALL: Specifies that this PTF function should occur on all of the service requesters that are defined for this service provider.

NONE: The service requesters are identified in a node list object.

NETATR: The remote network identifier of the service requester is the same as the network identifier of this system.

remote-network-identifier: Specify the remote network identifier of the service requester.

Element 2: Control Point

This element is not valid when the network identifier is *ALL, *SELECT, or *NONE.

remote-control-point: Specify the remote control point name of the service requester.

Optional Parameters

MAXDST

Specifies the maximum number of kilobytes that can be sent to each service requester.

NOMAX: Sends any number of kilobytes.

***SRVRQS:** The number of kilobytes that are specified for each service in the service requester entry information is used. Use the Work with Service Requesters (WRKSRVRQS) command to display the current value.

***SRVPVDA:** The number of kilobytes specified on the maximum distribution service provider attribute is used. Use the Display Service Provider Attribute (DSPSRVPVDA) command to determine the current value.

maximum-number-of-kilobytes: Specify the maximum number of kilobytes that can be distributed. Specify a value ranging from 1 through 5000 KB.

SNDRQT
Specifies if requisite PTFs should be added to the list.

***YES:** The PTF and all of its requisites are sent.

***NO:** The specific PTF or list of PTFs are sent. No requisites are sent.

Note: When using SNDRQT(*NO), all requisite PTFs must be applied on the service requester. The PTFs are applied in the order listed on this command.

CHECK
Specifies that each service requester is checked to determine its unique set of required PTFs.

***YES:** Each service requester is checked to determine the required set of PTFs.

***NO:** The service requesters are not checked to determine whether the PTFs are required. All PTFs are sent to all specified service requesters.

PTFPART
Specifies whether PTFs and cover letters should be sent.

***ALL:** Sends the PTF and its associated cover letter. If a cover letter is not found for a PTF, the cover letter is not sent.

***PTF:** Sends only the PTF.

***CVRLTR:** Sends only the cover letter.

APY Specifies the extent of change when the PTFs are applied. The apply occurs when all the PTFs have been successfully received by the service requester.

***NONE:** Do not apply PTFs on the service requester’s system.

***TEMP:** Apply the PTFs temporarily on the service requester’s system.

***PERM:** Apply the PTFs permanently on the service requester’s system.

DLYAPY
Specifies how PTFs are applied. Apply immediate PTFs at the time the activity runs on the specified system or later during the next IPL.

***NO:** Applies an immediate PTF at the time the activity runs. If the PTF is marked delayed, it is not applied until the next unattended IPL.

***YES:** Applies both immediate or delayed PTFs during the next unattended IPL.

NODL Specifies the node list object name that contains a list of service requesters where this PTF function is to be performed.

***NONE:** The service requesters are identified in the DESTSRVRQS parameter.

The possible library values are:

***LIBL:** Searches all of the libraries in the user and system portions of the job's library list for the node list object.

***CURLIB:** Searches for the node list object in the library for the current job.

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

node-list-name: Specify the node list name which contains the list of systems where the PTF function is to be performed.

RMTAPYTIME

Specifies the date and time when the apply can occur on the service requester. The current date and time values and next date are determined, when the apply request is processed on this system, based on this system's date and time.

Element 1: Time Zone

***LCLSYS:** Specifies the remote start time in the time zone of this system.

***MGDSYS:** Specifies the remote start time in the time zone of the service requester's system.

Element 2: Start After Time

***CURRENT:** Starts the apply on the service requester system at any time on or after the apply request is processed on this system.

start-after-time: Specify the time when the apply can be started on the service requester. The time can be entered as 4 or 6 digits, such as hhmm or hhmmss where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

Element 3: Start After Date

***CURRENT:** Starts the apply on the service requester system on any date on or after the date the apply request is processed on this system.

***NEXT:** Starts the apply on the service requester system on any date after the date the apply request is processed on this system.

start-after-date: Specify the date when the apply can start on the service requester system. The date must be specified in the job date format.

Note: The special values *CURRENT and *NEXT cannot be specified for the date and time when the time zone value *MGDSYS is specified.

IPL Specifies if an IPL of the service requester's system should occur.

***NO:** An IPL of the service requester system should not occur.

***YES:** An IPL of the service requester system should occur after all the PTFs are successfully applied on the service requester's system.

RMTIPLTIME

Specifies the date and time when this IPL can occur on the service requester. The current date and time values and next date are determined, when this IPL request is processed on this system, based on this system date and time.

Element 1: Time Zone

***LCLSYS:** Specifies the remote start time in the time zone of this system.

***MGDSYS:** Specifies the remote start time in the time zone of the service requester's system.

Element 2: Start After Time

***CURRENT:** Starts the IPL on the service requester system at any time on or after this IPL request is processed on this system.

start-after-time: Specify the time when this IPL can be started on the service requester. The time can be entered as 4 or 6 digits, such as hhmm or hhmmss where hh = hours, mm = minutes, and ss = seconds. Seconds are optional. The time can be specified with or without a separator. With a time separator, specify a string of 5 or 8 digits (hh:mm or hh:mm:ss).

Element 3: Start After Date

***CURRENT:** Starts the IPL on the service requester system on any date on or after the date the IPL request is processed on this system.

***NEXT:** Starts the IPL on the service requester system on any date after the date this IPL request is processed on this system.

start-after-date: Specify the date when the IPL can start on the service requester system. The date must be specified in the job date format.

Note: The special values *CURRENT and *NEXT cannot be specified for the date and time if the time zone value *MGDSYS is specified.

PWRDWNOPT

Specifies how to manage the end of active jobs during IPL process at the service requester system. This parameter is valid only if IPL(*YES) is specified.

***CNTRLD:** All jobs end when the specified time in the Power down delay (PWRDWNDLY) parameter expires. Any current jobs that are running will perform an end of batch processing.

***IMMED:** Jobs end immediately.

PWRDWNDLY

This parameter is valid only if IPL(*YES) and if PWRDWNOPT(*CNTRLD) are specified. Specifies the amount of time in seconds that the system allows a controlled end to be performed. If the end of job routines are not finished in the specified power down delay, any remaining jobs are ended immediately.

3600: The amount of time in which to complete a controlled end of processing is limited to 3600 seconds.

delay-time: Specify the maximum amount of delay time, in seconds, in which a controlled end can be performed. Valid values range from 1 through 65535 (65,535 seconds).

Examples for SNDPTF

Example 1: Sending a PTF and Requisites

```
SNDPTF  PTFID((1234567) (1234600)) DESTSRVRQS(*ALL)
   MAXDST(*SRVRQS)
```

This command sends a PTF and requisites to one or more selected service requesters.

Example 2: Sending a PTF to a Specific Service Requester

```
SNDPTF  PTFID(1234567)  DESTSRVRQS(*NETATR CHICAGO1)
   SNDRQT(*NO)
```

This command distributes PTF 1234567 to a service requester with a control point name of CHICAGO1, if the PTF is not already there. The size of the PTF package does not have an effect on the distribution and no requisites are sent with the PTF.

Example 3: Sending a PTF and Its Requisites to Two Service Requesters

```
SNDPTF  PTFID(1234567)  DESTSRVRQS((*NETATR CHICAGO1)
   (*NETATR CHICAGO2))    CHECK(*NO)
```

This command distributes PTF 1234567 and its requisites to two service requesters with control point names of CHICAGO1 and CHICAGO2. The service requesters are not checked before the PTFs are distributed to determine whether PTF 1234567 is already there. If any of the PTFs exist on the service requesters, they are replaced.

Example 4: Selecting One or More Service Requesters as Destination

```
SNDPTF  PTFID(1234567)   DESTSRVRQS(*SELECT)
```

This command shows a display that allows the user to select one or more service requesters as the destination for PTF 1234567 and its requisites.

Example 5: Sending and Temporarily Applying One PTF

```
SNDPTF  PTFID(1234567 *SPT *ONLY)
  DESTSRVRQS(*NETATR CHICAGO1)   SNDRQT(*NO)   CHECK(*NO)
  PTFPART(*PTF)   APY(*TEMP)
  RMTAPYTIME((*MGDSYS ('10:00:00' '10/31/02'))
  IPL(*YES)   RMTIPLTIME((*MGDSYS ('01:00:00' '11/01/02'))
```

This command sends and temporarily applies one PTF on a specific requester, schedules when the PTF is applied, and schedules the IPL.

This example loads and applies the specified PTFs at 10:00 a.m. on October 31, 2002 if all the PTFs were received successfully. If the PTFs have been received and applied successfully, the IPL on the remote system starts at 1:00 a.m. on November 1, 2002.

Example 6: Sending and Applying a Permanent PTF

```
SNDPTF  PTFID(1234567 *SPT *ONLY)
  DESTSRVRQS((*NETATR CHICAGO1)  (*NETATR CHICAGO2))
  PTFPART(*PTF)  APY(*PERM)
  RMTAPYTIME((*MGDSYS ('10:00:00' '10/31/02'))
  IPL(*YES)   RMTIPLTIME((*MGDSYS ('01:00:00' '11/01/02'))
```

This command sends and permanently applies one PTF to more than one service requester. This schedules the application of the PTF and the IPL of the system. The scheduled time is the time at the location of the service requester or the remote system. The command sends requisites and checks the service requester for the availability of the PTFs.

This example applies the specified PTFs on the service requester starting at 10:00 a.m. on October 31, 2002, if all the PTFs were received successfully. If the PTFs have been received and applied successfully, the remote system IPL starts at 1:00 a.m. on November 1, 2002.

Example 7: Doing an IPL after Sending and Applying a PTF

```
SNDPTF  PTFID(1234567 *SPT *ONLY)
  DESTSRVRQS(*SELECT)  PTFPART(*PTF)
  APY(*TEMP)  RMTAPYTIME((*MGDSYS (*CURRENT *CURRENT))
  IPL(*YES)   RMTIPLTIME((*LCLSYS (*CURENT 12/24/02))
  PWRDWNOPT(*CNTRLD) PWRDWNDLY(4800)
```

This command sends and temporarily applies one PTF to one or more selected service requesters. This command schedules the application of the PTF and the IPL of the system. The scheduled time is the time at the location of the service requester or the remote system. The command sends requisites and checks the service requester or the availability of the PTFs.

This example applies the specified PTF on the service requester starting at the current time on the current date. If the PTFs were received and applied successfully, the remote system IPL starts at the current time on December 24, 2002. The time specified for the IPL is in the time zone of the local system.

Error messages for SNDPTF

***ESCAPE Messages**

SMU107F
Error found processing &1 command.

SMU1080
Cannot specify *SELECT for service requester destination.

SMU109C
Errors detected while processing PTFs.

SMU109E
One or more PTFs could not be sent.

SMU109F
No PTFs sent.

***STATUS Messages**

SMU109A
Contacting &1.&2 to determine required PTFs.

«

SNDPTFORD (Send Program Temporary Fix Order) Command Description

SNDPTFORD Command syntax diagram

Purpose

» The Send Program Temporary Fix Order (SNDPTFORD) command allows the user to prepare an order for:

- Individual PTFs
- Cumulative PTF package
- PTF Group
- Summary information for available PTFs
- Preventive Service Planning (PSP) information

«

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

Required Parameter

PTFID Specifies the list of PTFs being ordered.

Note: If the PTF specified is for a product that is not installed or supported and CHKPTF is *NO, the PTF is sent without requisites.

Element 1: PTF Identifier

PTF-identifier: Specify the PTF identifier.

Some PTFs must be ordered individually or within a list of PTFs with the same prefix and not as part of a general list. Two PTF numbers are reserved and cannot be ordered.

The following chart shows special classes of PTFs and ordering restrictions.

PTF	Single	Group	Cover PTF	Reserved
Prefix	PTF	PTF	Letter	PTF ID
IInnnnn		X	X	
MF98nnn		X	X	
MF99nnn				X
SF95nnn				X
SF97vrm	X		X	
SF98nnn		X		X
SF99vrm	X			
SF99nnn		X		
SHnnnnn	X			

The following PTFs fall into the special classes:

IInnnnn
Information APAR

MF98nnn
Hardware preventive service planning information

MF99nnn
Reserved PTF

SF95nnn
Reserved PTF

SF97vrm
PTF summary

SF98nnn
Software preventive service planning information

SF99vrm
» Cumulative PTF package

«

SF99nnn
PTF group

SHnnnnn
Manufacturing, accounting and production information control system (MAPICS*) and construction management and accounting system (CMAS) PTFs

```
n is the PTF number

v is the version for the package requested
r is the release for the package requested
m is the modification level
  for the package requested
```

Element 2: Product

***ONLYPRD:** The PTF identifier is associated with only one product.

Note: If this value and ORDER(*REQUIRED) are specified, requisites are sent for only that product that is installed or supported on your system.

product-identifier: Specify the product identifier. If the PTF identifier is associated with more than one product, the PTF order is limited to the product specified.

Element 3: Release Level

***ONLYRLS:** The PTF identifier is associated with only one release.

Note: If this value and ORDER(*REQUIRED) are specified, requisites are sent for only that release level that is installed or supported on your system.

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

Optional Parameters

PTFPART

Specifies whether PTFs or cover letters are being ordered.

***ALL:** PTFs and cover letters are being ordered.

***CVRLTR:** Cover letters only are being ordered.

DELIVERY

Specifies how the PTFs are delivered.

***LINKONLY:** PTFs are delivered by the electronic customer support service link only.

***ANY:** PTFs are delivered by an any available method. The service link is used for most PTFs. PTFs that are too large for the service link are sent on the selected medium.

RMTCPNAME

Specifies the destination of the service provider to whom the service request is sent.

***IBMSRV:** The service request is sent to IBM service support.

***SELECT:** A list of service providers is shown so the user can select the destination for the service request.

remote-control-point-name: Specify the name of the control point that is the destination of the request.

RMTNETID

Specifies the remote name of the service provider's network.

***NETATR:** The service provider is in the local network.

remote-network-identifier: Specify the network name of the service provider to whom the request is being sent.

ORDER

Specifies the level of fixes that are being requested.

***REQUIRED:** The PTF ordered and its requisites are being requested.

***PTFID:** The specific PTF ordered is the one being requested. No requisites are sent.

REORDER

Specifies whether a PTF that is currently installed or ordered is ordered again. If ORDER(*REQUIRED) is specified, this parameter also applies to requisites.

***NO:** PTFs that are already on order or installed are not reordered.

***YES:** PTFs that are already installed or on order are reordered.

Note: A PTF is not reordered if a save file is available on the system.

CHKPTF

Specifies whether checking is performed on the service requester system to determine if PTFs are ordered based on whether or not the PTF product is installed or supported.

***NO:** The PTFs specified on the PTF identifier (PTFID) parameter are ordered even when the PTF product is not installed or supported on the service requester.

***YES:** The PTFs specified on the PTF identifier (PTFID) parameter are ordered only if the PTF product is installed or supported on the service requester.

Examples for SNDPTFORD

Example 1: Sending a Request

```
SNDPTFORD  PTFID((1234567) (1234600))
```

This command sends a request for PTF numbers 1234567 and 1234600.

Example 2: Sending a Request

```
SNDPTFORD   PTFID((1234567 5722SS1 V5R1M0))
  DELIVERY(*ANY)  ORDER(*REQUIRED)
```

This command sends a request for PTF 1234567 for the OS/400 product 5722-SS1 and the Version 5 Release 1 Modification 0. The PTF can be delivered by any available method. Any requisites are sent with the PTFs.

Example 3: Sending a Request

```
SNDPTFORD  PTFID((SF99360))
```

This command requests that the latest PTF cumulative package be sent for Version 5 Release 1 Modification 0.

Example 4: Sending a Request

```
SNDPTFORD PTFID((SF99893))
```

This command sends a request for PTF group number SF99893.

Error messages for SNDPTFORD

*ESCAPE Messages

CPF2702

Device description &1 not found.

CPF7C08

No support network connection.

CPF8C0C

Content of problem record &1 not valid.

CPF8C0E
Library QGPL not found.

CPF8C01
Cannot connect to IBM service system. One session allowed.

CPF8C07
A parameter is not valid.

CPF8C08
Cannot specify *SELECT for the control point name.

CPF8C09
&1 not defined as a service provider.

CPF8C16
Error occurred while processing request.

CPF8C17
Sign-on failed.

CPF8C18
No support network connection.

CPF8C19
Remote support application failed.

CPF8C2A
Cannot connect to IBM service system.

CPF8C24
Error occurred while processing request.

CPF8C27
Alternate load device not found.

CPF8C32
PTF order cannot be processed.

CPF8C99
PTF &2-&1 &3 not ordered.

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

SNDRPY (Send Reply) Command Description

SNDRPY Command syntax diagram

Purpose

The Send Reply (SNDRPY) command sends a reply message to the sender of an inquiry message. The message that is answered is the one having the specified message reference key that was received at the specified message queue. If the specified message queue is not allocated to the job in which this command is entered or to any other job, it is implicitly allocated by this command for the duration of the command.

Required Parameters

MSGKEY
Specifies the message reference key of the message that the reply answers.

MSGQ

Specifies the qualified name of the message queue that received the inquiry message being answered.

The name of the message 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.

message-queue-name: Specify the name of the message queue that received the inquiry message.

Optional Parameters

RPY Specifies the reply that the program sends as a response to the inquiry message.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used.

'reply-text': Specify the text (enclosed in apostrophes if it contains blanks or special characters) or a CL variable that contains the text that is sent as the program's reply to the inquiry message. The number of characters and their format are defined by the validity specifications given in the Add Message Description (ADDMSGD) command for the specified inquiry message. However, if no validity specifications are specified for replies in the ADDMSGD command, as many as 132 characters can be used in the reply text.

Coded Character Set Identifier (CCSID) Considerations

If the inquiry message that this reply is being sent to is an immediate message, the text supplied on the RPY parameter is assumed to be in the CCSID of the job running this command unless a coded character set identifier is supplied in the CCSID parameter. If the inquiry message that this reply is sent to is a predefined message, the text supplied in the RPY parameter is assumed to be 65535 and is not converted. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center.

RMV Specifies whether the inquiry message and its reply are removed from the specified message queue.

***YES:** The message and its reply are removed from the message queue when the reply is sent.

***NO:** The message and its reply are retained in the message queue. The inquiry message cannot be replied to more than once, but it can be received or displayed several times.

CCSID

Specifies the coded character set identifier (CCSID) that the supplied reply text is in. If the inquiry message that this reply is sent to is a predefined message, this parameter is ignored. Also, the text supplied on the RPY parameter is assumed 65535 and is not converted. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center.

***JOB:** The reply text is assumed to be in the CCSID of the job running this command.

***HEX:** The reply text is not converted. CCSID 65535 is used.

coded-character-set-identifier: Specify a valid CCSID in which you want your reply text to be considered in. Valid values range from 1 through 65535. This command will validate the CCSID. See the Globalization topic in the Information Center for a list of valid CCSID values.

Example for SNDRPY

```
SNDRPY   MSGKEY(&KEY)  MSGQ(SMITH)  RPY(YES)
```

This command sends a reply of YES to the message whose reference key is specified by &KEY, which was received at message queue SMITH. Because the reply contains only one word, the reply does not have to be enclosed in apostrophes.

Error messages for SNDRPY

*ESCAPE Messages

CPF2401
Not authorized to library &1.

CPF2403
Message queue &1 in &2 not found.

CPF2408
Not authorized to message queue &1.

CPF2410
Message key not found in message queue &1.

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

CPF2420
Reply already sent for inquiry or notify message.

CPF2422
Reply not valid.

CPF2432
Cannot send reply to message type other than *INQ or *NOTIFY.

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

CPF2460
Message queue &1 could not be extended.

CPF247E
CCSID &1 is not valid.

CPF2471
Length of field not valid.

CPF2477
Message queue &1 currently in use.

CPF2547
Damage to message file QCPFMSG.

CPF2548
Damage to message file &1 in &2.

CPF9830
Cannot assign library &1.

CPF9838
User profile storage limit exceeded.

SNDRPTJS (Send Reports using Job Scheduler) Command Description

Note: To use this command, you must have the 5722-JS1 (Job Scheduler for iSeries) licensed program installed.

SNDRPTJS Command syntax diagram

Purpose

The Send Reports using Job Scheduler (SNDRPTJS) command allows you to distribute reports to a specified report distribution ID as part of Job Scheduler's report distribution function. This command can be used outside of Job Scheduler in a user's control language program for distributing reports specified in the report distribution ID to the specified recipients.

Required Parameter

RPTDSTID
Specifies the name of the report distribution ID whose entries are used when distributing spooled files created by the specified job.

report-distribution-ID: Specify the report distribution ID.

Optional Parameter

JOB Specifies the name of the user job whose spooled files are being distributed.

This job identifier is a three-part qualified name. It can be specified several ways. For example:

job

user-name/job

job-number/user-name/job

***:** The job whose spooled files are being distributed is the job where this SNDRPTJS command is issued.

job-name: Specify the qualified name of the job whose spooled files are being distributed. If no job qualifier is given, all jobs currently in the system are searched for the job. If more than one job of the specified name is found, a qualified job must be specified.

user-name: Specify the name of the user of the job whose spooled files are being distributed.

job-number: Specify the number of the job whose spooled files are being distributed.

Example for SNDRPTJS

Example 1: Distributing Reports

```
SNDRPTJS RPTDSTID(GLEOM)
```

In this example a job is being distributed using report distribution ID GLEOM.

Error messages for SNDRPTJS

None

SNDSRVRQS (Send Service Request) Command Description

SNDSRVRQS Command syntax diagram

Purpose

The Send Service Request (SNDSRVRQS) command establishes a communications session and sends problem information to the service support system.

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

Required Parameters

ACTION

Specifies the type of requests processed during this session.

***PREPARED:** All records in the problem log with a status of PREPARED are sent to the remote service support system.

***OPENED:** All records in the problem log with a status of OPENED are sent to the remote service support system.

***TEST:** A test is done on the remote support function and the communications link to it.

Optional Parameters

RMTCPNAME

Specifies the service provider to whom the service request is sent. When ACTION(*PREPARED) is specified, only the problem log entries that have defined destinations are processed.

***IBMSRV:** The service request is sent to IBM service support.

***SELECT:** A list of service providers is shown so the user can select the destination for the service request.

remote-control-point-name: Specify the remote control point name of the service provider to whom the request is sent.

RMTNETID

Specifies the remote name of the service provider's network.

***NETATR:** The service provider is in the local network.

remote-network-identifier: Specify the network name of the service provider to whom the request is being sent.

Examples for SNDSRVRQS

Example 1

```
SNDSRVRQS  ACTION(*PREPARED)
```

This command establishes the communications link to the IBM service support system and sends all records in the problem log with the status prepared. The result of each problem log entry reported may be one of the following:

- PTFs sent to the system
- PTFs ordered from the code distribution center
- CE contacted automatically
- Service support center contacted automatically. The service support center representative will call you.
- Parts list

Example 2

```
SNDSRVRQS  ACTION(*OPENED)  RMTCPNAME(*SELECT)
```

This command allows the user to select a service provider from a list. The service provider will receive all records in the problem log with an opened status.

Error messages for SNDSRVRQS

*ESCAPE Messages

CPF2702
Device description &1 not found.

CPF7C08
No support network connection.

CPF8C0A
Data received from &1 not recognized.

CPF8C0B
Test request routed to different system than specified.

CPF8C0C
Content of problem record &1 not valid.

CPF8C0E
Library QGPL not found.

CPF8C0F
Error indicated in reply to request.

CPF8C01
Cannot connect to IBM service system. One session allowed.

CPF8C06
No problems in problem log can be sent.

CPF8C07
A parameter is not valid.

CPF8C08
Cannot specify *SELECT for the control point name.

CPF8C09
&1 not defined as a service provider.

CPF8C16
Error occurred while processing request.

CPF8C17
Sign-on failed.

CPF8C18
No support network connection.

CPF8C19
Remote support application failed.

CPF8C2A
Cannot connect to IBM service system.

CPF8C2B
Error indicated in reply to request.

CPF8C2D
Problem &1 cannot be sent.

CPF8C24
Error occurred while processing request.

CPF8C27
Alternate load device not found.

CPF9824
Not authorized to command &1 in library &2.

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

»

SNDSMGOBJ (Send System Manager Object) Command Description

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

SNDSMGOBJ Command syntax diagram

Purpose

The Send System Manager Object (SNDSMGOBJ) command provides the capability of sending an OS/400 object from the central site system to one or more managed systems.

Note: A change request is automatically submitted which can be viewed to determine the status of this command. A message is returned identifying the name of the change request.

Restrictions:

1. Only OS/400 program objects or file members such as *REXX, *CL, and *UNSPEC can be run.
2. The object cannot reside in the QTEMP library.
3. If a node list (NODL) value is specified, the node list can only contain entries that have a value of *SNA for the address type.

Required Parameters

OBJ Specifies the qualified name to be sent.

The possible library values are one of the following:

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

***CURLIB:** The current library for the job is used to locate the object.

library-name: Specify only the library named in this parameter is searched.

object name: Specify that characters A through Z, and 0 through 9 are used in the object names.

OBJTYPE
Specifies the object.

object-type: Specify the object type. If *FILEDATA object type is specified, a file member is sent without the file attributes.

Optional Parameters

MBR Specifies the OS/400 physical file member name. It is ignored unless the object type is *FILE or *FILEDATA.

<u>*ALL:</u> The entire file is sent. *ALL must be used if the file is not a physical file. *ALL cannot be used if RUNPGM(*YES) is selected or if *FILEDATA is selected.

***FIRST:** This is used only if the first member is sent. The member name is determined when the activity is run.

***LAST:** Only the last member is sent. The member name is determined when the activity is run by the date the member was added to the physical file.

member-name: Specify the name of the member to be sent.

DATATYPE
Specifies the data type of the member. This parameter needs to be specified only when sending a source file that runs on the managed system. DATATYPE is specified only when the object type is *FILEDATA.

<u>*UNSPEC:</u> Unspecified file member type. If the data type cannot be determined at the managed system, or if the name of the file where this member resides is QCLSRC, then the file member is treated as a CL batch input stream. If the source file is named QREXSRC, the file member is treated as a REXX procedure.

***CL:** The file member contains control language, in other words, an OS/400 CL batch input stream.

***REXX:** The file member contains a REXX procedure.

NODL Specifies the node list name contains a list of systems which are the destinations for the activities. This parameter cannot be specified if control point name (CPNAME) parameter is also specified.

<u>*NONE:</u> The systems on which this activity is performed are not identified by a node list. Individual control point names must be specified.

The possible library values are one of the following:

***LIBL:** All of the libraries in the user and system portions of the job's library list are searched for the node list object.

***CURLIB:** The current library for the job is used to locate the node list object.

library-name: Specify that only the library named in the parameter is searched.

node-list-name: Specify the node list object name containing the list of systems on which the activity is to be performed.

CPNAME
Specifies the managed systems APPN control point names on which this request is performed. Control point names cannot be specified if the node list (NODL) parameter is specified.

***NONE:** The systems on which this request is performed are not identified individually. A node list must be specified.

***NETATR:** The local system network ID is used. This is useful when the node specified is in the same network as the local system.

network-identifier: Specify the managed system APPN network identifier on which the request is performed.

control-point-name: Specify the managed system APPN control point name on which the request is performed.

TGTRLS
Specifies the release of the operating system on which you intend to use the object.

***CURRENT:** The object is used on the release of the operating system currently running on your system. If V5R2M0 is running on your system, *CURRENT means that you intend to use the object on a system with V5R2M0 installed. The object can also be used on a system with any later release of the operating system installed.

***PRV:** The object is intended for a system which is at the previous release level compared to the local system.

release-level: Specify the release level in the VxRxMx format. The object is used on a system with the specified release or with any later release of the operating system installed.

Valid values depend on the current version, release, and modification level and they change with each new release.

DTACPR
Specifies that the data is compressed when sent. This parameter is valid only if the object type *FILEDATA is specified. SNA compression character or blank is performed.

***NONE:** The file data is not compressed when sent.

***SNA:** The file data is compressed when sent.

RUNPGM
Specifies running the program when it arrives at the managed system. This parameter can only be specified if the object type is *PGM or if the object type is *FILEDATA and the data type is *CL or *REXX.

***NO:** The program is not run.

***YES:** Specify that the program is run as soon as it is received by the managed system.

REPLACE
Specifies that the object should be replaced if it already exists.

***NO:** An error is returned if the object already exists.

***YES:** The object is replaced if it already exists.

PARM Specifies the parameters to be passed when starting the program. Up to 20 parameters can be specified.

parameter: Specify a 1 to 253 character parameter. The prompt panel initially allows 50 characters to be entered. By entering an ampersand (&) in position 1, the field expands for larger parameters.

Examples for SNDSMGOBJ

Example 1: Sending a Library to Systems in a Node List

```
SNDSMGOBJ  OBJ(ACCTLIB)   OBJTYPE(*LIB)
  NODL(SYSLIB/SALESOFC)
```

Send a library ACCTLIB to 20 systems identified in node list SALESOFC.

Example 2: Sending a Program to All Systems in the Node List

```
SNDSMGOBJ  OBJ(SALESLIB/GENRPTS)   OBJTYPE(*PGM)
   NODL(SYSLIB/ALLSTORES)  RUNPGM(*YES)
   PARM(SALES ONE DAY)
```

Sends the program GENRPTS in library SYSLIB to all the systems listed in node list ALLSTORES in library SYSLIB. It then runs the program on each of those systems. Two parameters are passed as input to the program. The first has the value SALES and the second has the value ONE DAY.

Error messages for SNDSMGOBJ

*ESCAPE Messages

None «

SNDTCPSPLF (Send TCP/IP Spooled File) Command Description

SNDTCPSPLF Command syntax diagram

Purpose

The Send TCP/IP Spooled File (SNDTCPSPLF) command sends a spooled file to be printed on a remote system. The remote system must be running TCP/IP. The SNDTCPSPLF command is the iSeries version of the TCP/IP LPR (Line Printer Requester) command.

Required Parameters

RMTSYS
Specifies the remote system name to which the print request is sent.

To be successful, the name must be valid, and the system must be able to communicate with the local system. You can assign names to an internet address. Use the Work with TCP/IP host table entries option on the Configure TCP/IP menu (CFGTCP command). A remote name server also maps 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.

If this parameter is left blank, you are prompted for the INTNETADR parameter when the Enter key is pressed.

***INTNETADR:** The value of the INTNETADR parameter is used.

remote-system: Specify the remote system name to which the print request is sent.

PRTQ Specifies the name of the destination printer queue. For destination systems that are iSeries, this is the name of an output queue to which the spooled file is sent.

If a library name qualifier is not specified, the library list of the sending user ID on the destination system is searched. If not found or not authorized, QGPL/QPRINT is used as the default. The output queue is usually specified as library name/output queue name (for example, QGPL/OUTQ1). For destination systems that are not iSeries, this name is system-dependent. This name can be either the actual name of the device or the name of a printer queue for the device. The name used for the OS/2 operating system is the actual name of the device. Often, the name used for the UNIX operating system is the name of the printer queue for the device. This parameter is limited to 128 characters.

Note: On some systems, the printer queue name may be case sensitive. To preserve the lowercase letters in a printer queue name, enclose the printer queue name in apostrophes 'MyPrtQ'.

FILE Specifies the name of the spooled file that is sent to the specified system and printer. The file name is the same file name that is displayed with the Work with Spooled File (WRKSPLF) command.

Optional Parameters

JOB Specifies the name of the job that created the spooled file whose data records are to be sent.

***:** The job that issued this command is the job that created the spooled file.

job-name: Specify the qualified name of the job that created the spooled file.

user-name/job-name: Specify the user name and the name of the job that created the spooled file.

job-number/user-name/job-name: Specify the job number, the user name, and the name of the job that created the spooled file. The name of the job has a fully specified form of job-number/user-name/job-name. If no job number or user name is given, all jobs currently running on the system are searched for a match against the job name.

SPLNBR
Specifies the number of the spooled file from the job whose data records are to be sent.

***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, an error 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 having the specified file name whose data records are to be sent.

JOBSYSNAME
Specifies the name of 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, spooled file number, and creation date and time is used.

***ANY:** The job system name is not used to determine which spooled file is used. Use this value when the spooled file 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.

DESTTYP

Specifies the type of destination system to which the spooled file is being sent. This information allows the SNDTCPSPLF command to send the file in a format that is most likely to be received and processed correctly on the destination system. When sending to another iSeries, this parameter should be *AS400.

***OTHER:** The destination system is not an iSeries and is not using Print Services Facility/2.

***AS400:** The destination system is an iSeries. If TRANSFORM(*NO) is specified, a duplicate of the spooled file is created on the destination system. If the spooled file is *SCS or *AFPDS and TRANSFORM(*YES) is specified, the spooled file is transformed to ASCII and spooled as *USERASCII on the destination iSeries.

***PSF2:** The destination system is using Print Services Facility/2. This destination system can receive both ASCII data and Advanced Function Printer data stream (AFPDS) data. This means that in addition to sending ASCII data to a PSF/2, you can also send a spooled file of device type *AFPDS. When sending *AFPDS files, you must consider any external resources of the file. These resources, such as fonts and overlays, must also reside on the destination system to allow the file to print correctly.

TRANSFORM

Specifies whether to make use of the host print transform function to transform a SNA character string (SCS) or Advanced Function Printing data stream (AFPDS) into ASCII data. If you specify DESTTYP(*OTHER), you typically specify TRANSFORM(*YES). This is because most non-iSeries destination systems do not support printing of non-ASCII data. The non-iSeries destination systems do receive and correctly process ASCII data.

Note: If you specify TRANSFORM(*YES), you cannot use the USRDTATFM parameter.

***YES:** Use the host print transform.

***NO:** Do not use the host print transform.

MFRTYPMDL

Specifies the manufacturer, type, and model for a printer using the host print transform function.

This parameter is prompted only when TRANSFORM(*YES) is specified.

***IBM42011:** The IBM 4201-1 Proprinter is used.

manufacturer-type-model: Specify the manufacturer, type, and model for a printer using the host print transform function. Specify one of the values in Table 1 (page 257) that matches the device.

***WSCST:** The value of the WSCST parameter is used.

Table 1 (page 257) lists the printer values for the manufacturer, type, and model.

Table 1. Manufacturer Type and Model (MFRTYPMDL Parameter)***

*IBM2380	IBM 2380 Personal Printer Series II
	IBM 2380 Plus Printer
*IBM2381	IBM 2381 Personal Printer Series II,
	IBM 2381 Plus Printer
*IBM2390	IBM 2390 Personal Printer Series II
	IBM 2390 Plus Printer
*IBM2391	IBM 2391 Personal Printer Series II
	IBM 2391 Plus Printer
*IBM3112	IBM 3112 Page Printer
*IBM3116	IBM 3116 Page Printer
*IBM3130	IBM 3130 Advanced Function Printer
*IBM3812	IBM 3812 Pageprinter
*IBM3816	IBM 3816 Pageprinter
*IBM3912HP	IBM 3912 Page Printer (HP Mode)
*IBM3916HP	IBM 3916 Page Printer (HP Mode)
*IBM39302	IBM 39302 IBM 3930-02S Page Printer
	IBM 39302 IBM 3930-02D Page Printer
*IBM39303	IBM 39303 IBM 3930-03S Page Printer
	IBM 39303 IBM 3930-03D Page Printer
*IBM4019	IBM 4019 LaserPrinter
	IBM 4019E LaserPrinter E
*IBM4019HP	IBM 4019 LaserPrinter (HP** Mode)
	IBM 4019E LaserPrinter E (HP Mode)
*IBM4029	IBM 4029-010 LaserPrinter 5E
	IBM 4029-020 LaserPrinter 6
	IBM 4029-030 LaserPrinter 10
	IBM 4029-040 LaserPrinter 10L

*IBM4029HP	IBM 4029-010 LaserPrinter 5E (HP Mode)
	IBM 4029-020 LaserPrinter 6 (HP Mode)
	IBM 4029-030 LaserPrinter 10 (HP Mode)
	IBM 4029-040 LaserPrinter 10L (HP Mode)
*IBM4037	IBM 4037 5E Printer
*IBM4039HP	IBM LaserPrinter 4039-10D (HP Mode)
	IBM LaserPrinter 4039-10D Plus (HP Mode)
	IBM LaserPrinter 4039-10R (HP Mode)
	IBM LaserPrinter 4039-10R Plus (HP Mode)
	IBM LaserPrinter 4039-12R (HP Mode)
	IBM LaserPrinter 4039-12R Plus (HP Mode)
	IBM LaserPrinter 4039-12L (HP Mode)
	IBM LaserPrinter 4039-12L Plus (HP Mode)
	IBM LaserPrinter 4039-16L (HP Mode)
	IBM LaserPrinter 4039-16L Plus (HP Mode)
*IBM4070	IBM 4070 IJ
*IBM4070EP	IBM 4070 IJ (Epson** Mode)
*IBM4072	IBM 4072 ExecJet*
*IBM4076	IBM 4076 ExecJet* II Printer (HP Mode)
*IBM42011	IBM 4201-1 Proprinter*
*IBM42012	IBM 4201-2 Proprinter II
*IBM42013	IBM 4201-3 Proprinter III
*IBM42021	IBM 4202-1 Proprinter XL
*IBM42022	IBM 4202-2 Proprinter II XL
*IBM42023	IBM 4202-3 Proprinter III XL
*IBM42071	IBM 4207-1 Proprinter X24
*IBM42072	IBM 4207-2 Proprinter X24E
*IBM42081	IBM 4208-1 Proprinter XL24
*IBM42082	IBM 4208-2 Proprinter XL24E
*IBM4212	IBM 4212 Proprinter 24P
*IBM4216	IBM 4216-10 Personal Pageprinter
*IBM4226	IBM 4226-302 Printer
*IBM4230	IBM 4230-4S3 Printer (IBM Mode)
	IBM 4230-4I3 Printer (IBM Mode)
*IBM4232	IBM 4232-302 Printer (IBM Mode)
*IBM4244MAN	IBM4244 Printer (single tractor or manual feed selection)
*IBM4244ASF	IBM4244 Printer (single tractor feed and ASF)

*IBM4244DUAL	IBM4244 Printer (dual tractor feeds)
*IBM4247MAN	IBM4247 Printer (single formfeed or manual selection)
*IBM4247DUAL	IBM4247 Printer (dual tractor feeds)
*IBM4247ASF	IBM4247 Printer (single tractor feed and ASF)
*IBM4312	IBM Network Printer 12
*IBM4317	IBM Network Printer 17
*IBM4320	IBM InfoPrint 20
*IBM4324	IBM Network Printer 24
*IBM4332	IBM InfoPrint 32
*IBM47121	IBM 4712-1 Transaction Printer
*IBM47122	IBM 4712-2 Transaction Printer
*IBM47221	IBM 4722-1 Document Printer
*IBM47222	IBM 4722-2 Document Printer
*IBM4770	IBM 4770 InkJet Transaction Printer
*IBM5152	IBM 5152 Graphics Printer
*IBM5201	IBM 5201-2 Quietwriter*
*IBM5202	IBM 5202-1 Quietwriter III
*IBM5204	IBM 5204-1 Quickwriter*
*IBM5216	IBM 5216 Wheelprinter

*IBM5575	IBM 5579-H02 Printer
	IBM 5579-K02 Printer
	IBM 5577-T02 Printer
	IBM 5579-S02 Printer
	IBM 5577-K02 Printer
	IBM 5577-J02 Printer
	IBM 5577-G02 Printer
	IBM 5577-H02 Printer
	IBM 5577-F02 Printer
	IBM 5577-B02 Printer
	IBM 5575-H02 Printer
	IBM 5575-F02 Printer (with SBCS Cartridge)
	IBM 5575-B02 Printer (with SBCS Cartridge)
	IBM 5573-K02 Printer
	IBM 5573-J02 Printer
	IBM 5573-H02 Printer
	IBM 5573-G02 Printer
	IBM 5572-B02 Printer
	IBM 5417-011 Printer
	IBM 5407-011 Printer
	IBM 5327-011 Printer
	IBM 4208-502 Printer
*IBMPAGES	IBM 5589-H01 Printer
	IBM 5588-H02 Printer
	IBM 5587-H01 Printer
	IBM 5586-H02 Printer
	IBM 5585-H01 Printer
	IBM 5584-K02 Printer
	IBM 5584-H02 Printer
	IBM 5584-G02 Printer
*IBMPAGES300	IBM Network Printer 12 (with PAGES feature)
	IBM Network Printer 17 (with PAGES feature)
	IBM Network Printer 24 (with PAGES feature)
*IBM6400	IBM 6400 Printers (IBM Mode)
*IBM6400EP	IBM 6400 Printers (Epson Mode)
*IBM6404	IBM 6404 Printers (IBM Mode)
*IBM6404EP	IBM 6404 Printers (Epson Mode)

*IBM6408	IBM 6408-A00 Printer (IBM Mode) IBM 6408-CTA Printer (IBM Mode)
*IBM6408EP	IBM 6408-A00 Printer (Epson Mode) IBM 6408-CTA Printer (Epson Mode)
*IBM6412	IBM 6412-A00 Printer (IBM Mode) IBM 6412-CTA Printer (IBM Mode)
*IBM6412EP	IBM 6412-A00 Printer (Epson Mode) IBM 6412-CTA Printer (Epson Mode)
*CANLIPS3	Canon LIPS3 DBCS Printers
*CPQPM15	COMPAQ** PageMark 15 (HP Mode)
*CPQPM20	COMPAQ PageMark 20 (HP Mode)
*HPDBCS	HP LaserJet-compatible printers for Double Byte Character Set (DBCS) input.
*HPII	HP LaserJet** Series II
*HPIID	HP LaserJet IID
*HPIIP	HP LaserJet IIP
*HPIII	HP LaserJet III
*HPIIID	HP LaserJet IIID
*HPIIIP	HP LaserJet IIIP
*HPIIISI	HP LaserJet IIISi
*HP1200C	HP DeskJet 1200C
*HP1600C	HP DeskJet 1600C
*HP310	HP DeskJet 310
*HP320	HP DeskJet 320
*HP4	HP LaserJet 4
*HP4000	HP LaserJet 4000 series
*HP5	HP LaserJet 5 series
*HP500	HP DeskJet 500
*HP520	HP DeskJet 520
*HP540	HP DeskJet 540
*HP550C	HP DeskJet 550C
*HP560C	HP DeskJet 560C
*HP5000	HP LaserJet 5000 series
*HP5SI	HP LaserJet 5Si
*HP6	HP LaserJet 6 series
*HP8000	HP LaserJet 8000 series
*HPCOLORLJ	HP Color LaserJet 5

*HPPAINT	HP PaintJet
	HP PaintJet XL
	HP PaintJet XL300
*LEXOPTRA	Lexmark Optra Family (HP Mode)
*LEXOPTRAC	Lexmark Optra C Color Printer
*LEXOPTRAN	Lexmark Optra N Printer
*LEXOPTRAS	Lexmark Optra S Printer family
*LEXOPTRASC	Lexmark Optra SC Color Printer
	Lexmark Optra Color 1200 Printer
*LEX2380	Lexmark Forms Printer 2380 Plus
*LEX2381	Lexmark Forms Printer 2381 Plus
*LEX2390	Lexmark Forms Printer 2390 Plus
*LEX2391	Lexmark Forms Printer 2391 Plus
*LEX4227	Lexmark 4227 Forms Printer
*ESCPDBCS	Epson ESC/P DBCS Printers
*EPAP2250	Epson ActionPrinter 2250
*EPAP3250	Epson ActionPrinter 3250
*EPAP5000	Epson ActionPrinter 5000
*EPAP5500	Epson ActionPrinter 5500
*EPDFX5000	Epson DFX-5000
*EPDFX8000	Epson DFX-8000
*EPFX850	Epson FX-850
*EPFX870	Epson FX-870
*EPFX1170	Epson FX-1170
*EPLQ570	Epson LQ-570
*EPLQ860	Epson LQ-860
*EPLQ870	Epson LQ-870
*EPLQ1070	Epson LQ-1070
*EPLQ1170	Epson LQ-1170
*EPLQ510	Epson LQ-510
*EPLQ2550	Epson LQ-2550
*EPLX810	Epson LX-810
*EPSQ870	Epson SQ-870
*EPSQ1170	Epson SQ-1170
*EPEPL7000	Epson EPL-7000
*EPEPL8000	Epson EPL-8000

*NECPR201	NEC** PC-PR101 NEC** PC-PR201
*NECP2	NEC** P2 Pinwriter
*NECP2200	NEC P2200 Pinwriter
*NECP2200XE	NEC P2200 XE Pinwriter
*NECP5200	NEC P5200 Pinwriter
*NECP5300	NEC P5300 Pinwriter
*NECP6200	NEC P6200 Pinwriter
*NECP6300	NEC P6300 Pinwriter
*NONE	Printer supports page-descriptor language generated by the CVTIMG API. Note: Spoolfiles with device type of *SCS or *AFPDS cannot be processed by the Host Print Transform function for these printers.
*OKI184IBM	Okidata** Microline 184 Turbo (IBM Mode)
*OKI320IBM	Okidata Microline 320 (IBM Mode)
*OKI321IBM	Okidata Microline 321 (IBM Mode)
*OKI390IBM	Okidata Microline 390 Plus (IBM Mode)
*OKI391IBM	Okidata Microline 391 Plus (IBM Mode)
*OKI393IBM	Okidata Microline 393 Plus (IBM Mode)
*OKI590IBM	Okidata Microline 590 (IBM Mode)
*OKI591IBM	Okidata Microline 591 (IBM Mode)
*OKI400	Okidata OL400 LED Page Printer
*OKI800	Okidata OL800 LED Page Printer
*OKI810	Okidata OL810 LED Page Printer
*OKI820	Okidata OL820 LED Page Printer
*OKI3410	Okidata Pacemark 3410
*PAN1123EP	Panasonic** KX-P1123 (Epson Mode)
*PAN1124EP	Panasonic KX-P1124 (Epson Mode)
*PAN1124IEP	Panasonic KX-P1124i (Epson Mode)
*PAN1180EP	Panasonic KX-P1180 (Epson Mode)
*PAN1180IEP	Panasonic KX-P1180i (Epson Mode)
*PAN1191EP	Panasonic KX-P1191 (Epson Mode)
*PAN1624EP	Panasonic KX-P1624 (Epson Mode)
*PAN1654EP	Panasonic KX-P1654 (Epson Mode)
*PAN1695EP	Panasonic KX-P1695 (Epson Mode)
*PAN2123EP	Panasonic KX-P2123 (Epson Mode)
*PAN2124EP	Panasonic KX-P2124 (Epson Mode)
*PAN2180EP	Panasonic KX-P2180 (Epson Mode)
*PAN2624EP	Panasonic KX-P2624 (Epson Mode)

*PAN4410HP	Panasonic KX-P4410 (HP Mode)
*PAN4420HP	Panasonic KX-P4420 (HP Mode)
*PAN4430HP	Panasonic KX-P4430 (HP Mode)
*PAN4450IHP	Panasonic KX-P4450i (HP Mode)
*PAN4451HP	Panasonic KX-P4451 (HP Mode)
*XRX4215MRP	Xerox** 4215/MRP (HP Mode)
*XRX4219MRP	Xerox 4219/MRP (HP Mode)
*XRX4220MRP	Xerox 4220/MRP (HP Mode)
*XRX4230MRP	Xerox 4230/MRP (HP Mode)
*XRX4235	Xerox 4235 LaserPrinting (HP Mode)
*XRX4700II	Xerox 4700 II Color Document Printer (HP Mode)
*WSCST	Printer is not listed
*WSCSTLETTER	Printer is not listed (letter-sized paper)
*WSCSTLEGAL	Printer is not listed (legal-sized paper)
*WSCSTLEDGER	Printer is not listed (ledger-sized paper)
*WSCSTEXECUTIVE	Printer is not listed (executive-sized paper)
*WSCSTA3	Printer is not listed (A3-sized paper)
*WSCSTA4	Printer is not listed (A4-sized paper)
*WSCSTA5	Printer is not listed (A5-sized paper)
*WSCSTB4	Printer is not listed (B4-sized paper)
*WSCSTB5	Printer is not listed (B5-sized paper)
*WSCSTCONT80	Printer is not listed (8-inch continuous forms)
*WSCSTCONT132	Printer is not listed (13.2 inch continuous forms)
*WSCSTNONE	Printer is not listed (paper size not specified)

USRDTATFM

Specifies the qualified name of the program used to transform the spooled file data. User data transform programs must be written to the Writer Transform Exit Program interface. For information on this interface, see the Application Program Interfaces (APIs) topic in the Information Center.

***NONE:** No data transform program is specified.

The name of the data transform 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.

data-transform-program-name: Data transform program used to transform the spooled file data.

INTNETADR

Specifies the internet address of the remote system to which the print request is sent.

This parameter is accessed by entering *INTNETADR for the RMTSYS parameter.

internet-address: Specify the internet address of the remote system. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

WSCST

Specifies an object that consists of a table of attributes used to customize a given ASCII device, such as a workstation or printer. Character presentation, font specifications, and control key sequences are examples of characteristics that can be customized. This parameter is prompted only when TRANSFORM(*YES) is specified.

If the WSCST parameter is used in conjunction with the MFRTYPMDL parameter, information that is specified in the workstation customization object takes precedence over the MFRTYPMDL parameter.

***NONE:** Do not specify a workstation customizing object.

The name of the WSCST 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.

workstation-customizing-object: Specify a valid workstation customizing object that was created with the Create Work Station Customizing Object (CRTWSCST) command.

DLTSPLF

Specifies whether or not to delete the specified spooled file after it has been successfully sent. No guarantee can be made as to the correct printing of the sent file. Only correct transmission can be verified.

***NO:** Do not delete the spooled file after it has been successfully sent.

***YES:** Delete the spooled file after it has been successfully sent.

DESTOPT

Specifies the destination-dependent options. These options may be specific to a particular implementation of an LPD print server. When this parameter is not specified, the SNDTCPSPLF command sends default options that are common to all LPD print servers. The options specified by this parameter are inserted into the control file generated by the SNDTCPSPLF command, one option per line, and sent to the specified LPD print server. The options specified by this parameter must have the following format:

```
"option1 option2 option3...optionN"
```

where each option is separated from the next by one or more blanks.

Options typically have a format of <control characters><value>. An example is

```
 "-oTITLE=LPR"
```

where “-o” are the control characters and “TITLE=LPR” is the value. SNDTCPSPLF examines the first character of each option to determine if it is a reserved control character. The following control characters cannot be specified as their values are determined by SNDTCPSPLF. If any of the following control characters are specified, the option is ignored and a warning message is generated.

```
C        T
H        U
I        W
L        1
M        2
N        3
P        4
S        k
```

For example “Class=ASCII” would not be allowed because the “C” control character is reserved for use by SNDTCPSPLF. However, “-Class=ASCII” would be permitted.

The control character “X” is allowed, but the value of “AS/400” used with it is not permitted. For example, “XMyLPD” can be specified, but “XAS/400” cannot be. The option “XAS/400” is reserved for use by the SNDTCPSPLF command. It is used to indicate to another iSeries that the request is coming from an iSeries.

The option “XAIX” controls how multiple copies are sent to the printer. This option specifies that the data and control files are to be sent multiple times, once for each copy. The default is to send the data and control files once, with multiple print commands in the control file.

The following print “filters” are supported by SNDTCPSPLF (see section 7 of RFC 1179):

```
c    n
d    p
f    r
g    t
l    v
```

If one of these characters is specified as the first character in a destination option, the character is used on the “print file” command in the control file. Only the first occurrence of one of these characters in this position is used in the control file. Subsequent use of these characters in this position is ignored. If none of these options is specified, the default character “l” is used on the “print file” command, which prints the file “as is” without filtering out any control characters.

If “J” is specified as the first character of a destination option, the remaining characters in the option are used to override the default job name for the banner page (see RFC 1179, section 7.4). For instance, to specify a job name of /nff, specify:

```
 DESTOPT('J/nff')
```

Only the first occurrence of this option is used to override the job name. If this option is not specified, the spooled file name is used as the job name (the default).

You can enable an auto-queue option by using the destination dependent option XAUTOQ on the SNDTCPSPLF command (or on the CRTOUTQ or CHGOUTQ commands). When you use the auto-queue option, files are sent as before unless the remote system times out during the transform. If this happens, the transformed spool files are sent back to the same output queue using the AS/400 LPD server. When a remote writer is running, it will then send the transformed spooled file to the remote system.

destination-dependent-options: Specify the destination-dependent options.

SEPPAGE
Specifies whether or not a separator page is requested when the spooled file is printed on the remote system.

Note: A separator page may not be printed in all cases. Printing of separator pages depends on whether the LPD implementation on the receiving system honors the separator page request. Some LPD implementations ignore the SEPPAGE parameter and print or do not print a separator page by default.

***YES:** Request a separator page.

***NO:** Do not request a separator page.

Examples for SNDTCPSPLF

The following examples assume that the spooled file is the only one with that name within the job under which you are currently running when making the request.

Example 1: Sending a File to Another iSeries

```
SNDTCPSPLF RMTSYS(SYS1)
  PRTQ(DISTRIB/LAMP) FILE(SALES)
  DESTTYP(*AS400) TRANSFORM(*NO)
```

This command sends the spooled file named SALES to the output queue LAMP in the library DISTRIB on the remote iSeries system SYS1. An exact duplicate of the spooled file is created on the destination system because it is an iSeries system-to-iSeries system transfer as specified on the DESTTYP parameter. The spooled file is not transformed.SNDTCPSPLF

Example 2: Sending a File to a System of Type *OTHER

```
SNDTCPSPLF RMTSYS(SUN) PRTQ(MOON)
  FILE(STARS) DESTTYP(*OTHER)
  TRANSFORM(*YES) MFRTYPMDL(*HPIIID)
```

This command sends the spooled file STARS to the printer queue MOON on the remote system SUN. If the spooled file data stream is SCS, the file is transformed to the correct ASCII format for a Hewlett-Packard IIID Printer.

Example 3: Sending a File to a System of Type *OTHER

```
SNDTCPSPLF RMTSYS(*INTNETADR)
  PRTQ(MOON) FILE(STARS) DESTTYP(*OTHER)
  TRANSFORM(*YES) MFRTYPMDL(*WSCST)
  WSCST(USERLIB/QWPMYPRT)
  INTNETADR('9.5.255.128')
```

This command sends the spooled file STARS to the printer queue MOON on the remote system with the Internet address 9.5.255.128. If the spooled file data stream is SCS, the file is transformed to the correct ASCII format for a custom printer definition. The workstation customizing object (created with the CRTWSCST command) can be found in file QWPMYPRT, library USERLIB.

Example 4: Sending a File to a System of Type *PSF2

```
SNDTCPSPLF RMTSYS(OS2SRVR) PRTQ(PSFQ)
  FILE(JOBLOG) DESTTYP(*PSF2)
  TRANSFORM(*YES) MFRTYPMDL(*IBM5202)
```

This command sends the spooled file JOBLOG to the printer queue PSFQ on the remote system OS2SRVR. If the spooled file data stream is SCS, the file is transformed to the correct ASCII format for an IBM 5202 Printer.

Example 5: Using a Transform Program Other than the Host Print Transform

```
SNDTCPSPLF RMTSYS(OS2SRVR) PRTQ(PSFQ)
  FILE(JOBLOG) DESTTYP(*PSF2)
  TRANSFORM(*NO) USRDTATFM(library/program)
```

This command sends the spooled file JOBLOG to the printer queue PSFQ on the remote system OS2SRVR. The file is transformed using the program specified with the USRDTATFM parameter.

Error messages for SNDTCPSPLF

***ESCAPE Messages**

TCP3701
Send request failed for spooled file &1.

TCP3702
Not authorized to spooled file &1.

TCP3703
Send request failed. Spooled file &1 open.

TCP3704
Send request failed. Spooled file &1 not spooled print file.

TCP3705
Spooled file &1 not transformed.

TCP3706
Spooled file &1 not transformed.

TCP3715
Not authorized to delete spooled file &1. The spooled file was sent.

SNDTIEF (Send Technical Information Exchange File) Command Description

SNDTIEF Command syntax diagram

Purpose

The Send Technical Information Exchange File (SNDTIEF) command sends a specified file to the remote support network.

Required Parameters

FILE Specifies the physical file to be sent to the remote support network.

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

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

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

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

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

MBR Specifies the name of the database file member transmitted to the remote support network.

The possible member names are:

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

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

TOUSRID
Specifies the ID of the user who receives the file.

TONODE
Specifies the name of the system that receives the file.

Optional Parameters

TOFILE
Specifies the name of the file on the receiving system.

***FILE:** The name of the database file on the receiving system is the same as that specified on the FILE parameter.

file-name: Specify the name of the database file on the receiving system.

TYPE Specifies the type of contents of the file.

***OTHER:** The content of the file is not specified.

***SYSTPLD:** The file contains system configuration descriptions.

PTY Specifies the transmission priority that is assigned to the file.

2: The file has normal priority.

1: The file has the highest priority.

Example for SNDTIEF

```
SNDTIEF  FILE(QSYS/MYTOPO)  TOUSRID(INFSERV)
  TONODE(INFTIE)  TOFILE(ACMETOPO)
```

This command sends a file named MYTOPO from library QSYS to TIE. It is held in a mailbox for user INFSERV on system INFTIE. When it is received by the user, it is named ACMETOPO.

Error messages for SNDTIEF

None

SNDUSRMSG (Send User Message) Command Description

SNDUSRMSG Command syntax diagram

Purpose

The Send User Message (SNDUSRMSG) command is used by a program to send a message to a message queue and, optionally, to receive a reply to that message. The message sent using this command can be either an immediate message or a predefined message and it can be sent to a display station user either in an interactive job queue or in a specific message queue. For inquiry messages, a CL variable can be specified to receive the reply value, and the program using this command will wait for a response.

This command uses a combination of parameters available on the Send Program Message (SNDPGMMSG) and Receive Message (RCVMSG) commands to allow a program to send and receive messages by using a single command. Also, the SNDUSRMSG command provides validity checking and uppercase translation for replies to inquiry messages.

Restrictions:

1. This command is valid only in CL programs.
2. The SNDUSRMSG command allows a message of up to 512 characters of first-level message text to be sent. However, if the message is sent to an external message queue (*EXT) in an interactive job, only 76 characters are shown on the Display Program Messages display. If the message is sent to a user's, work station's, or system operator's message queue, the Display Message (DSPMSG) command allows all 512 characters to be displayed.

Required Parameters

MSG Specifies the message text that is sent. A maximum of 512 characters can be specified. The string must be enclosed in apostrophes if special characters (including blanks) are used. If a value is specified on this parameter, the MSGID, MSGF, and MSGDTA parameters cannot be specified.

Coded Character Set Identifier (CCSID) Considerations

The text supplied on the MSG parameter is assumed to be in the CCSID of the job running this command unless a coded character set identifier is supplied in the CCSID parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center.

MSGID
Specifies the message identifier of a predefined message sent by the program to a message queue. If a value is specified for the MSGID parameter, a value must also be specified for the MSGF parameter, while the MSG parameter cannot be specified.

MSGF Specifies the qualified name of the message file that contains the predefined message (specified in the MSGID parameter) being sent. The MSGF parameter is valid only if a predefined message is being sent.

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

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

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

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

message-file-name: Specify the name of the message file to use.

Optional Parameters

MSGDTA

Specifies the character string or the CL variable that contains a character string used as the message data in the predefined message. A character string that is specified contains one or more substitution values that are used in place of the substitution variables that were defined in the message's text when the message was defined. The MSGDTA parameter is valid only if a predefined message is being sent. For more information, refer to the description of the MSGDTA parameter of the Send Program Message (SNDPGMMSG) command.

***NONE:** No message data is specified for the predefined message.

'message-data': Specify the character string that gives the substitution values in the specified predefined message that is sent, or specify the name of the variable that contains the character string.

Coded Character Set Identifier (CCSID) Considerations

The text supplied on the MSGDTA parameter that corresponds to the *CCHAR type field is assumed to be in the CCSID of the job running this command unless the coded character set identifier is supplied in the CCSID parameter. All other text supplied on the MSGDTA parameter is assumed to be 65535 and is not converted. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center. For more information about the *CCHAR type field, see the Add Message Description (ADDMSGD) command.

VALUES

Specifies a list of valid replies to an inquiry message sent by this command. No more than 20 replies can be specified in the list. If the reply to the inquiry message does not match one of the specified values, an error message is sent to the reply's sender and the inquiry message is sent again. This value is valid only when a value is specified for the MSGRPY parameter.

Translation of the reply, if requested (on the TRNTBL parameter), will occur before verification of a match between the reply and the specified value. If TRNTBL(*LIBL/QSYSTRNTBL) is used to translate the reply values from lowercase to uppercase and lowercase characters still appear in a specified value, there is no match.

Variables can be used in the list of replies. This parameter is valid only if MSGTYPE(*INQ) is specified.

***NONE:** No predefined replies to inquiry messages are given. Any reply to an inquiry message is valid.

allowable-values: Specify no more than 20 values that are compared to replies received for inquiry messages sent by this command. The maximum length of each value is 32 characters.

DFT Specifies the value used as the reply to an inquiry message (sent by this command) if the inquiry message is sent to a message queue that is in default delivery mode, or if for any other reason the default reply is sent. The value specified here as the default reply does not have to match an entry in the list of valid replies specified in the VALUES parameter. The default reply value is not translated to uppercase or validity-checked. This parameter is valid only if a value is specified for the MSGRPY parameter.

***MSGDFT:** The default value defined in the message description of the message ID (specified on the MSGID parameter) is used. If no message ID is specified, the default value is *N.

'default-reply-value': Specify the reply (enclosed in apostrophes) which is used as the default reply. If a predefined message is being sent, this default-reply-value overrides any default reply specified in the message description.

MSGTYPE
Specifies the type of message sent. Only inquiry and informational messages can be specified.

***INQ:** An inquiry message is sent and the message queue receiving the message must reply to it.

***INFO:** An informational message is sent.

TOUSR
Specifies that the message is sent to the message queue specified in the user profile for the user named on this parameter. This parameter cannot be used if TOMSGQ is specified.

***SYSOPR:** The message is sent to the system operator's message queue, QSYS/QSYSOPR.

***REQUESTER:** The message is sent to the user profile's message queue for interactive jobs or to the system operator's message queue (QSYS/QSYSOPR) for batch jobs.

user-profile-name: Specify the user profile name of the user to whom the message is sent.

TOMSGQ
Specifies the qualified name of the message queue to which the message is sent. This parameter cannot be used if TOUSR is specified.

***:** In an interactive job, the message is sent to the external message queue (*EXT). In a batch job, the message is sent to the system operator's (QSYS/QSYSOPR) message queue.

***SYSOPR:** The message is sent to the system operator's message queue, QSYS/QSYSOPR.

***EXT:** The message is sent to the job's external message queue. For batch job inquiry messages, the default reply is always received. Messages sent to this queue can be 512 characters in length, but only 76 characters are displayed on the Display Program Messages display.

The name of the message 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.

message-queue-name: Specify the name of the message queue that is to receive the message being sent.

MSGRPY
Specifies the CL character variable (of up to 132 characters) that contains the reply received in response to an inquiry message. This parameter is valid only if MSGTYPE(*INQ) is specified. If the reply is longer than the variable, it is truncated; if it is shorter, it is padded to the right with blanks. This command will wait for a response to the inquiry message even if no value is coded for this parameter when MSGTYPE(*INQ) is specified. If no value is coded for this parameter when MSGTYPE(*INQ) is specified, the command awaits a response from the operator in reply to the inquiry message.

TRNTBL
Specifies, if the reply value is being translated, the qualified name of the translate table used.

The name of the translate 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.

QSYSTRNTBL: The translate table named QSYSTRNTBL (found in library *LIBL) is used to translate the reply value.

The IBM-supplied translate table QSYSTRNTBL table translates, for the English language only, all lowercase characters in the range of X'81' to X'A9' to uppercase characters. All other characters are not translated.

To use a different translate table, use the Create Table (CRTTBL) command and specify that particular table for this parameter.

table-name: Specify the name of the translate table used to translate the message reply.

Other Single Values

***NONE:** The reply is not being translated.

CCSID

Specifies the coded character set identifier (CCSID) that the message text supplied on the MSG (message) or MSGDTA (message data) parameters is in. If a message identifier is specified, the text supplied by the MSGDTA (message data) parameter that corresponds to the *CCHAR type field is assumed to be in the CCSID supplied by this parameter. The data supplied that does not correspond to the *CCHAR type field is assumed to be 65535 and is not converted. For more information about the *CCHAR type field see the Add Message Description (ADDMSGD) command.

If no message identifier is specified, the text supplied by the message parameter is assumed to be in the CCSID supplied by this parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the Information Center.

***JOB:** The text supplied by the MSGDTA (message data) or MSG (message) parameters is assumed to be in the CCSID of the job running this command.

***HEX:** The text supplied by the MSGDTA (message data) or MSG (message) parameters is not converted. CCSID 65535 is used.

coded-character-set-identifier: Specify a valid CCSID in which you want your text specified on the MSG (message) or MSGDTA (message data) parameter to be considered in. Valid values range from 1 through 65535. This command validates the CCSID. See the Globalization topic in the Information Center for a list of valid CCSID values.

Examples for SNDUSRMSG

Example 1: Message Requiring Specific Reply

```
SNDUSRMSG  MSG('Data has been verified.  Do you
  want to update the master files (Y or N)?')
  TOMSGQ(*) VALUES(Y N) DFT(N) MSGRPY(&REPLY)
```

This command sends an inquiry message to the display station operator (if it is used in an interactive job) or to the system operator (if it is used in a batch job). The valid replies are Y and N, and any other reply is rejected. The reply is returned in the variable &REPLY. The default translation table, QSYSTRNTBL, is used to translate the reply to uppercase characters.

Example 2: Message Requiring Any Reply

```
SNDUSRMSG  MSG('Enter any response to this
  message when ready to continue with updates.')
  TOMSGQ(WS01)
```

This command sends an inquiry message to a specific message queue. Any reply is valid. Because the purpose of this example is simply to wait, no CL variable is provided to receive the reply.

Example 3: Sending an Information Message

```
SNDUSRMSG  MSGID(USR0150)  TOUSR(FRED)
  MSGF(QGPL/USRMSGF)
  MSGDTA(&ACCTNO)  MSGTYPE(*INFO)
```

This command sends a predefined message as an information message to the message queue specified in the user profile of the specified user (FRED). The message data provided is included in the message.

Error messages for SNDUSRMSG

***ESCAPE Messages**

CPF247E
CCSID &1 is not valid.

CPF2559
Error occurred in SNDUSRMSG command.

SNDRCVF (Send/Receive File) Command Description

SNDRCVF Command syntax diagram

Purpose

The Send/Receive File (SNDRCVF) command is used by a control language (CL) program to send data to and receive data from a device that is being used interactively by a user. The data is passed between the program in which the SNDRCVF command is used and the display device identified in the command. The data is passed using the display device file that was declared in the program. (A Declare File (DCLF) command included in the source used to compile the program was used to declare the file.) The data for each send/receive operation is passed as one record in a format identified by the RCDFMT parameter of this command (the format is defined in the data description specifications (DDS)). One CL variable is used for each field of the record format to pass the data. The CL variables used (including DDS indicators) are declared implicitly.

Of the record formats specified in the DCLF command, only one can be specified in each SNDRCVF command. If the device file has not been opened, it is opened by this command. The file and record format specified in this command can be overridden by an Override with Display File (OVRDSPF) command if that command is entered before the file is opened. However, care should be taken that the fields in the overriding record format correspond to the CL variables declared in the program.

Restriction: This command is valid only for display files within a CL program. It cannot be used with database files.

Optional Parameters

DEV Specifies the name of the display device to which the CL program's data is sent and from which the user's data is received. A CL variable can be specified on this parameter so that the device name can be changed without changing the command.

***FILE:** The data is sent to and received from the device associated with the device file (the device file that was declared in the FILE parameter of the DCLF command). If more than one device name is specified in the device file, *FILE cannot be specified.

device-name: Specify the name of the device or the name of the CL variable that contains the name of the device to which the CL program sends data and from which it receives data. If a CL variable name is used in this parameter, only one SNDRCVF command is needed in the program to receive data from several devices.

RCDFMT
Specifies the name of the record format that is used to pass the data between the CL program and the user. The format contains all the fields in the record. This parameter must be coded with a record format name if there is more than one record format in the device file; *FILE cannot be coded if there is more than one. SNDRCVF ignores the INVITE DDS keyword.

***FILE:** Only one record format is in the device file, which is used to send the data to and receive the data from the user.

record-format-name: Specify the name of the record format in which the data is sent to and received from the user. A CL variable name cannot be used here to specify the record format name.

WAIT Specifies whether the CL program waits to receive the data from the user's device or continues to process the commands that follow this SNDRCVF command. If WAIT(*NO) is specified, the program must issue a WAIT command later in the program to complete the input operation.

Note: A CL variable cannot be coded on this parameter.

***YES:** The program waits until the input operation from the device is completed before the next command is processed.

***NO:** The program does not wait for the input data; it continues to process commands until a WAIT command is reached later in the program.

Examples for SNDRCVF

Example 1

```
DCLF   FILE(MENU1)
  *
  *
  *
SNDRCVF
```

This command sends and receives user data by way of the device file MENU1. Only one record format exists in the file. The device used is specified in the file.

Example 2

```
DCLF   FILE(SCR)  RCDFMT(REC8)
  *
  *
  *
SNDRCVF   RCDFMT(REC8)
```

The CL program sends data to a user and receives data for the user who is using the device named in the device file (*FILE is assumed because DEV is not specified). The data is passed in the format specified by REC8 record format in the device file named SCR. The CL program waits for the user data before continuing.

Example 3

```
DCLF   FILE(DF1)  RCDFMT(REC8)
  *
  *
  *
SNDRCVF   DEV(&DN)  RCDFMT(REC8)  WAIT(*NO)
  *
  *
  *
WAIT   DEV(&DN)
```

This command sends and receives user data by way of the device file named DF1. Using the record format REC8, the CL program passes data between itself and the user who is at the device named in the variable &DN, but it does not wait for a response to come back. If the program sends and receives data from several devices, the same SNDRCVF command can be used. Only the device specified by &DN for the DEV parameter must be changed. A WAIT command for each device must be issued later in the program to ensure that all the devices respond.

Error messages for SNDRCVF

*ESCAPE Messages

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

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

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

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

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

CPF0886
Record contains a data field that is not valid.

CPF0887
Data available from previous input request.

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

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

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

SETASPGRP (Set ASP Group) Command Description

SETASPGRP Command syntax diagram

Purpose

The Set Auxiliary Storage Pool Group (SETASPGRP) command sets the auxiliary storage pool (ASP) group for the current thread. Additionally, this command allows you to change the libraries in the library list for the current thread. If an ASP group had already been set, this command will remove the old ASP group from the current thread and set the specified ASP group for the current thread. Once the specified ASP group has been set for the current thread, all libraries in the independent ASPs in the ASP group are accessible and objects in those libraries can be referenced using regular library-qualified object name syntax. The libraries in the independent ASPs in the specified ASP group plus the libraries in the system ASP (ASP number 1) and basic user ASPs (ASP numbers 2-32) form the *library name space* for the thread. All libraries in the library list need to be in the new library name space or the library list is not changed and the new ASP group is not set.

Restrictions:

- You must have *USE authority to all ASP device descriptions in the ASP group and to all the specified libraries in the library list before the library name space and the library list are changed. If you are not authorized to an ASP device description or to one of the libraries, the ASP group will not be set and the library list will not be changed.
- When *CURUSR is specfied for the ASPGRP or USRLIBL parameter, you must have *READ authority to the job description (JOBD) listed in your user profile and *EXECUTE authority to the library containing that JOBD.

Required Parameter

ASPGRP

Specifies the name of the auxiliary storage pool (ASP) group to set for the current thread. The ASP group name is the name of the *primary* ASP device within the ASP group.

auxiliary-storage-pool-group-name: Specify the name of the *primary* ASP in the ASP group to be set for the current thread. All libraries from all ASPs in this ASP group will be included in the library name space.

***CURUSR:** The ASP group is set to the value defined for the *Initial ASP group* in the default job description of the user profile that the thread is currently running under.

***NONE:** Specify for the current thread to have no ASP group. The library name space will not include libraries from any ASP group. Only the libraries in the system ASP and any basic user ASPs will be in the library name space.

Optional Parameters

SYSLIBL

Specifies the system part of the library list for the thread in which the command is entered.

***CURSYSBAS:** The libraries in the system part of the library list of the current thread that are found in the system ASP (ASP number 1) or any configured basic user ASP (ASP numbers 2-32) are used as the new system part of the library list.

***SYSVAL:** The system part of the library list is set from the current value of system value QSYSLIBL.

CURLIB

Specifies the library to be used in the current library entry of the library list for the thread. If *CURUSR or a library name is specified and the library cannot be found in the new library name space, an error message is sent and the library list and ASP group are not changed.

***CURSYSBAS:** The library name in the current library entry of the library list is used as the new current library if the library is found in the system ASP (ASP number 1) or any configured basic user ASP (ASP numbers 2-32). If the library name in the current entry is not found in the system ASP or any basic user ASP, the current library entry will be removed from the library list.

***CURUSR:** The current library is set to the value defined for *Current library* in the user profile that the thread is currently running under.

***CRTDFT:** Changes the library list to remove any name from the current library entry. If objects are created specifying *CURLIB for the library name, library QGPL will be used.

library-name: Specify the name of the library that replaces the current library entry in the library list.

USRLIBL

Specifies the libraries that are placed in the user part of the library list. If *CURUSR or a list of library names is specified and any of these libraries cannot be found in the new library name space, an error message is sent and the library list and ASP group are not changed.

***CURSYSBAS:** The libraries in the user part of the library list of the current thread that are found in the system ASP (ASP number 1) or any configured basic user ASP (ASP numbers 2-32) are used as the new user part of the library list.

***CURUSR:** The user part of the library list for the thread is set to the value defined for the *Initial library list* in the default job description of the user profile that the thread is currently running under.

***SYSVAL:** The user part of the library list is set from the current value of system value QUSRLIBL.

***NONE:** Changes the user part of the library list to remove all library names.

library-name: Specify the names of the libraries to be used as the user part of the library list, in the order in which they are to be searched. The number of libraries that can be specified ranges from 1 through 250.

Examples for SETASPGRP

Example 1: Set New ASP Group

```
SETASPGRP   ASPGRP(WAREHOUSE1) SYSLIBL(*SYSVAL)
      CURLIB(*CURUSR) USRLIBL(*CURSYSBAS)
```

This command will set the ASP group for the thread in which the command runs to be WAREHOUSE1. This will change the library name space for the thread to include all libraries in any of the independent ASPs in the ASP group identified by the independent ASP device named WAREHOUSE1. The system part of the library list will be set from system value QSYSLIBL. The current library entry of the library list will be set from the *Current library* value defined in the user profile that the thread is currently running under. The user part of the library list will be set using the current user part of the library list and removing any libraries that are not found in the system ASP or configured basic user ASPs.

Example 2: Set to No ASP Group

```
SETASPGRP   ASPGRP(*NONE) SYSLIBL(*CURSYSBAS)
      CURLIB(*CRTDFT) USRLIBL(*NONE)
```

This command will remove any ASP group for the thread in which the command runs. This will change the library name space for the thread to include only those libraries in the system ASP (ASP number 1) and basic user ASPs (ASP numbers 2-32). The system part of the library list will be set using the current system part of the library list and removing any libraries that are not found in the system ASP or configured basic user ASPs. The current library entry of the library list will be changed to be empty which will cause library QGPL to be used as the current library. The user part of the library list will be changed to be empty.

Error messages for SETASPGRP

***ESCAPE Messages**

CPFB8E9
ASP group &1 not set for thread &2.

«

SETATNPGM (Set Attention Program) Command Description

SETATNPGM Command syntax diagram

Purpose

The Set Attention Program (SETATNPGM) command causes the specified program to become the attention key handler at the current recursion level in the job running the command. If the ATTN key handler's status is SET(*ON), the specified program is called when the ATTN key is pressed. No parameters are passed to the ATTN key handler when it is called. The attention handling program runs in the same process with the same job attributes, overrides, and group authorities as the program that issued the SETATNPGM command. However, program adopted authority is not carried over.

This command is call-oriented; that is, a SETATNPGM command issued at one recursion level causes the attention handling program to be in effect at the current recursion level as well as deeper recursion levels, until another SETATNPGM command occurs which changes the attention handling program. More information is in the Work Management book.

Pressing the ATTN key under the following conditions does not call the ATTN key handler (that is, it appears as if SET(*OFF) were the status of the ATTN key):

- When the input inhibited light at the work station is on
- At the system request menu or any of its options
- At the break message display and the display message display
- While the OS/400 system is already calling the attention handler
- Some OS/400 system functions (3270 Emulation, 5250 Passthru) do not allow the ATTN key to call the attention handling program

Note:

Caution is necessary when defining an attention handling program. Attention handling is similar to system request option 1 (transfer to secondary job), but there are special considerations for its use. When a transfer to secondary job occurs, processing continues in another job; however, when the ATTN key is pressed, the ATTN key handler runs in the same job. The program in progress when the ATTN key is pressed is not protected by any locks held. This means that the attention handler can get a lock on any object, even if a program it interrupted already had an exclusive lock. Be aware that read from invited devices operation could time-out during the time that the attention key handling program is running. Therefore, if a time-out were to complete in the program in progress while the attention key handling program is running, whatever action taken as a result of that time-out occurs upon return to the program in progress.

For example, if the WAITRCD value in the file is set to 60 seconds, the program is set to exit if a key is not pressed in 1 minute, and the attention key program is invoked and runs longer than 1 minute, the program exits when it returns from the attention key handler.

However, caution should be used because a check for data available is made before checking to see that the time-out completes. If a key is pressed immediately after leaving the attention key handler, data could be available which could complete the read from invited devices and the time-out is not checked. This could cause unexpected results.

Applications that perform Get-No-Wait operations can be interrupted at any point where the ATTN key is pressed. Attention handlers therefore should:

- Use simple functions like a menu which transfers to a group job or secondary job.
- Avoid referring to objects or functions that may be in use when the ATTN key is pressed.
- Avoid calling nonrecursive functions (many non-OS/400 system functions) that are going to be interrupted by the ATTN key. (This is especially true of high-level languages and utilities, because they are not designed to be used recursively.)

Restriction: To use this command the program specified must exist and you must have object operational authority or one of the data authorities to it. You must also have read authority to the program’s library.

Required Parameter

PGM Specifies the qualified name of the program to be the ATTN key handler at this recursion level.

***CURRENT:** The program name of the ATTN key handler currently in effect is used as the value of this parameter. An error does not occur if an ATTN key handler is not currently in effect.

***PRVINVLVL:** The ATTN key handler in effect at the previous recursion level is reinstated as the ATTN key handler at this recursion level. The SET parameter is not allowed if this special value is specified, because the SET status of the previous recursion level is also reinstated. This option is used when a program has specified an attention program and wants to revert back to a previous level.

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 program that is the ATTN key handler.

Optional Parameter

SET Specifies whether the ATTN key handler indicated in the program (PGM) parameter should be called when the ATTN key is pressed. This parameter is not allowed if PGM(*PRVINVLVL) is specified.

***ON:** The ATTN key handler specified in the program (PGM) parameter is called when the ATTN key is pressed.

***OFF:** The ATTN key handler specified in the program (PGM) parameter is not called when the ATTN key is pressed.

Examples for SETATNPGM

Example 1: Setting the ATTN Key Handler

```
SETATNPGM  PGM(QGPL/ATTN)  SET(*ON)
```

This command causes the program QGPL/ATTN to become the ATTN key handler. Because SET(*ON) is specified, the program is called when the ATTN key is pressed.

Example 2: Setting the Attention Key Off

```
SETATNPGM  PGM(*CURRENT)  SET(*OFF)
```

The current attention handling program has its status changed to SET(*OFF). Because the status is SET(*OFF) when the ATTN key is pressed, the attention handling program is not called.

Example 3: Previous Recursion-Level Support

```
SETATNPGM  PGM(*PRVINVLVL)
```

The attention handling program and status that was in effect at the previous recursion level is reinstated at this recursion level. If no attention handler is in effect, after this command is run nothing happens when the ATTN key is pressed.

Example 4: Emulating the System Request Key

```
SETATNPGM  PGM(QWSSYSRQ)
```

The system-supplied program QWSSYSRQ will be called when the ATTN key is pressed. This system program allows the ATTN key to act as a system request key by showing the system request menu on the display when the ATTN key is pressed.

Error messages for SETATNPGM

***ESCAPE Messages**

CPF1318
Attention key program &1 in &2 not set.

SETCSTDTA (Set Customization Data) Command Description

SETCSTDTA Command syntax diagram

Purpose

The Set Customization Data (SETCSTDTA) command can be used to copy and delete information in order to customize OS/400 Graphical Operations support. For example, work area contents and mouse double-click actions can be customized and copied.

By using this command, administrators do not need to manually set up each OS/400 GraphicOps user. The administrators can set up users by modeling one user and patterning all subsequent users after the model user. The administrators can also reset user customization so they can restart the customization process if needed.

Required Parameters

USRPRF
Specifies the user profile to receive the customization data.

user-profile-name: Specify the names of the user profiles to reset or to which to copy. A maximum of 300 user profile names can be specified.

VALUE
Specifies the value to which the customization information is set.

***NONE:** The customization data for the user profile name is removed.

user-profile-name: Specify the name of the user profile from which to copy customization information.

Optional Parameter

REPLACE
Specifies whether to replace customization data for the user profile name specified on the USRPRF parameter.

***NO:** The customization data is not replaced. An exception message is issued.

***YES:** The customization data for the user profile name is replaced.

Examples for SETCSTDTA

Example 1: Copying Customization Data to a User Profile

```
SETCSTDTA   USRPRF(TOM)  VALUE(TIM)
```

This command copies OS/400 GraphicOps customization data associated with the user profile name TIM to the new user profile name TOM.

Example 2: Replacing Customization Data

```
SETCSTDTA   USRPRF(TOM)  VALUE(MODEL1)  REPLACE(*YES)
```

This command copies OS/400 GraphicOps customization data associated with the model user profile name MODEL1 to the user profile name TOM and replaces any customization data associated with the user profile name TOM.

Example 3: Removing Customization Data

```
SETCSTDTA  USRPRF(TOM) VALUE(*NONE)
```

This command removes all OS/400 GraphicOps customization data associated with user profile name TOM.

Error messages for SETCSTDTA

***ESCAPE Messages**

CPF2204
User profile &1 not found.

CPF2217
Not authorized to user profile &1.

GUI0085
&1 does not have customization data.

GUI0087
Some user profiles did not have customization data copied successfully.

GUI0091
Some user profiles did not have customization data deleted successfully.

GUI0117
&1 is not valid for subset key.

SETDEPJS (Set Dependent Job using Job Scheduler) Command Description

Note: To use this command, you must have the 5722-JS1 (Job Scheduler for iSeries) licensed program installed.

SETDEPJS Command syntax diagram

Purpose

The Set Dependent Job using Job Scheduler (SETDEPJS) command allows you to change the completion flag from *YES to *NO or vice versa for a predecessor job in a predecessor/successor relationship. The completion flag indicates whether a job has processed. This command allows you to indicate that a job has or has not processed so that a dependent job can be submitted or omitted from submission.

Required Parameter

PREDJOB
Specifies the name of the predecessor job schedule entry that you want to change. 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: Predecessor job

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

Element 2: Group

***NONE:** The job is not a member of a group.

group-name: Specify the group in which the job resides that you want to change the completion flag.

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.

Optional Parameters

SUCCJOB

Specifies the name of the successor job schedule entry that you want to change. You must specify a job and alternately can specify a group to which the job belongs as well as a sequence number for the job within the group.

Element 1: Successor job

***ALL:** Change the completion flag for all jobs on which this job is dependent.

successor-job-name: Specify the successor job whose predecessor job completion flag you are changing.

Element 2: Group

***NONE:** The job is not a member of a group.

group-name: Specify the name of the group whose predecessor job completion flag you are changing.

Element 3: Group sequence

***NONE:** The job does not have a sequence number.

group-sequence-number: Specify the sequence number of the successor job.

COMPLETE

Specifies whether you want to change the completion flag for a predecessor job from *YES to *NO or vice versa.

***YES:** Change the completion flag to *YES.

***NO:** Change the completion flag to *NO.

Examples for SETDEPJS

Example 1: Changing a Predecessor Job

```
SETDEPJS PREDJOB(JOB10) SUCCJOB(JOB11)
```

In the example, JOB10 has been changed to processed, thus freeing JOB11 to run as a successor job to JOB10.

Example 2: Triggering end of week processing

```
SETDEPJS PREDJOB(EOD) SUCCJOB(EOW) +
COMPLETE(*NO)
```

In this example an end of day (EOD) job that runs every week day at 5:00 p.m. is set up. A dependent (successor) job for end of week (EOW) processing could also be set up that is scheduled to run on Friday after the EOD job is completed. A third job called EOWCHECK (end of week check) could be set up whose only purpose is to change the completion flag on the EOD job from *YES to *NO. This EOWCHECK job would run prior to 5:00 p.m. on Friday. The reason you would want to do this is to allow the EOW job to run on Friday after the EOD job is completed. This command would be used in the EOMCHECK job.

Error messages for SETDEPJS

None

SETKBDMAP (Set Keyboard Map) Command Description

SETKBDMAP Command syntax diagram

Purpose

The Set Keyboard Map (SETKBDMAP) command allows the user to override the PA (Program Attention) and PF (Program Function) key assignment defaults provided by the iSeries 400.

This command assigns the specified F-to-PF map to the device where the command was entered (if it is a 3270 display station device) or to the 3270 display station specified if the user has authority to that device.

Optional Parameters

DEV Specifies a valid 3270 display station that is assigned this keyboard mapping function.

***REQUESTER:** This mapping is assigned to the device where the command is entered.

device-name: Specify a device other than the device named in the *REQUESTER value. The user must have allocation authority to the specified device. In a program environment, the user should either acquire or allocate the specified device before entering this command.

Key Sequence
Each valid key or key sequence is a separate parameter which, except for restrictions noted below, may be assigned any function. If the user does not specify a function for a particular key or key sequence, the default value becomes the function for that key or key sequence. Following is a list of functions and restrictions:

***ATTN** Attention

***BCKSPC**
Record Backspace

***CLEAR**
Clear Screen

***DOWN**
Roll Down

***DSPATR**
Display Imbedded Attributes

***F1-*F24**
F1 through F24 Function Keys

***HELP**
5250 Help

***HLP3270**
3270 Help Text (Display Active Keyboard Map)

***NONE**
No Assignment

***PRINT**
Print Screen

***RESET**
Error reset

***SAME**
No Change

***SYSREQ**
System Request

***TEST** Test Request

***UP** Roll up

Notes:

1. *HELP, *HLP3270, and *RESET must be assigned to PF1 through PF12, PA1/PF1 through PF12, or PA2/PF1 through PF12. Because these functions are required for 3270 display station device support to function properly, and not all keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless that function is also assigned to PF1 through PF12, PA1/PF1 through PF12, or PA2/PF1 through PF12.
2. It is recommended that *F1 and *SYSREQ be assigned to PF1 through PF12, PA1/PF1 through PF12, or PA2/PF1 through PF12.
3. The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

Example for SETKBDMAP

```
SETKBDMAP   PF1(*F1)  PF2(*F2)  PF3(*F3)
  PF4(*F4)  PF5(*HLP3270)  PF9(*HELP)
```

This command reassigns the keyboard primarily for an application that makes frequent use of the 5250 CF keys F1, F2, F3, F4. All other PF key sequences are set to the default shown on the command prompt. The above command is started in the program that started the application (thus tailoring the display station to whatever application is run).

Error messages for SETKBDMAP

***ESCAPE Messages**

CPF8701
Specified device &1 not a 3270 device type.

CPF8702
&1 function key not correctly defined.

CPF8703
Device &1 not ready.

SETMEDBRM (Set Media Controls Using BRM) Command Description

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

SAVMEDBRM Command syntax diagram

Purpose

The Set Media Controls using BRM (SETMEDBRM) command has two distinct parts, a set of input controls and a set of output controls.

Input controls pertain only to media from which data is being retrieved or read.

Output controls start with the MEDCLS parameter, continue through the TEXT parameter, and pertain to media on which data is being written. For example, if you plan to perform a CPYTOTAP operation, you can use the SETMEDBRM command beforehand to assign a move policy and retention period for the volumes that are used.

The controls you specify with SETMEDBRM remain in effect for your job until they are reset or otherwise changed by another SETMEDBRM command. To see control values that are currently in effect, type the SETMEDBRM command on a command line and press F4.

Examples for SETMEDBRM

Example 1: Using Input Controls

```
SETMEDBRM LABEL(*ANY 1) ALWCNV(*YES)
```

In this example any file with a version number 1 greater than the most current version is read. If ASCII data is found, convert it to EBCDIC.

Example 2: Using Output Controls

```
SETMEDBRM MEDCLS(QIC1000) MOVPCY(OFFSITE)
RET(*DATE '3/3/99')
```

In this example output operations will use media class QIC1000, a move policy of OFFSITE and a retention date of 3/3/99 will be assigned to any volumes produced while these output controls are in effect.

Error messages for SETMEDBRM

None

SETOBJACC (Set Object Access) Command Description

SETOBJACC Command syntax diagram

Purpose

The Set Object Access (SETOBJACC) command temporarily changes the speed of access to an object by bringing the object into a main storage pool or purging it from all main storage pools. An object can be kept resident in main storage by selecting a pool for the object that has available space and does not have jobs associated with it. Repeated use of the command can cause a set of objects to be resident in a main storage pool.

Required Parameters

OBJ Specifies the qualified name of the object to be brought into or purged from main storage.

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

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

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

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

***ALL:** All libraries on the system 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.

object-name: Specify the name of the object.

OBJTYPE
Specifies the type of object to be brought into or purged from main storage.

***FILE:** The object is a file.

***PGM:** The object is a program.

POOL Specifies whether the object is brought into or purged from main storage.

***JOB:** The object is brought into the pool associated with the job.

***BASE:** The object is brought into the base pool.

***SHRPOOLn:** The object is brought into a general-purpose shared pool. Valid pool values range from 1 through 10.

***PURGE:** The object is purged from all pools.

Element 1: Subsystem

subsystem: Specify a subsystem name.

Element 2: Pool Identifier

pool-identifier: Specify a subsystem pool identifier.

Optional Parameters

MBR Specifies the database file member to be brought into or purged from main storage.

***FIRST:** The first member is selected.

file-member-name: Specify the member name.

MBRDATA
Specifies the member data to be brought into or purged from main storage.

***BOTH:** All parts of the object are selected.

***ACCPTH:** The file member's access path is selected.

***DATA:** The file member's data is selected.

Example for SETOBJACC

```
SETOBJACC  OBJ(OBJA)  OBJTYPE(*PGM)  POOL(*JOB)
```

This command brings a program named OBJA to the pool associated with the job in which the command is run.

Error messages for SETOBJACC

***ESCAPE Messages**

CPF1858
The specified pool does not exist.

CPF1859
Use of an access path was requested but none exists.

CPF9855
File &1 in library &3 contains no members.

SETRTVBRM (Set Retrieve Controls for BRM) Command Description

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

SETRTVBRM Command syntax diagram

Purpose

The Set Retrieve Controls for BRM (SETRTVBRM) command can be used to change the way that retrieve operations work when performed by your job.

The controls you specify with SETRTVBRM remain in effect for the duration of your job or until they are reset or otherwise changed by another SETRTVBRM command. The controls you specify with this command are not kept when the job ends. To see control values that are currently in effect, type the SETRTVBRM command on a command line and press F4.

Note: You can retrieve data from a TSM server device by using this command. You can only specify one TSM device of *MEDCLS, which must select a TSM device. The TSM device selected can either be *APPC, which supports SNA, or *NET, which supports TCP/IP.

Example for SETRTVBRM

Example 1: Resetting Control Values for BRMS Retrieve Operations in Your Job

```
SETRTVBRM DEV(TAP01) RECALL(*NOTIFY *NOTIFY)
RTVAUT(*READ)
```

In this example, retrieve operations that occur during your job will use device TAP01, will simply notify you when they occur, and require that you only need read authority to be able to retrieve an object.

Error messages for SETRTVBRM

None

SETTAPCGY (Set Tape Category) Command Description

SETTAPCGY Command syntax diagram

Purpose

The Set Tape Category (SETTAPCGY) command sets the category for a resource in a specified media library device. The system automatically loads cartridges from the specified category in the order specified on the cartridge order (CTGORDER) parameter.

Required Parameters

MLB Specifies the name of the media library device for which the category is set.

OPTION
Specifies which option is set for the specified media library device.

***MOUNTED:** The category specified on the CGY parameter is considered mounted and any tape operation with VOL(*MOUNTED) specified uses the tape cartridges from the category that is mounted.

***DEMOUNTED:** The category specified on the CGY parameter is no longer mounted. The use of VOL(*MOUNTED) is not valid for the media library device.

***ASSIGN:** The mounted category session specified on the MNTID parameter is assigned to the job issuing the SETTAPCGY command. The mounted category session being assigned must have been previously mounted and released.

***RELEASE:** The mounted category session assigned to the job issuing the SETTAPCGY command is released and is available for another job to assign.

Optional Parameters

CGY Specifies the category to be mounted. This parameter is valid only when OPTION(*MOUNTED) is specified.

Element 1: Category name

***NOSHARE:** The cartridge identifiers cannot be shared with other systems that are attached to the same device. The cartridge identifiers are mounted in the order specified in the CTGORDER parameter.

***IPL:** The cartridge identifiers can be used for an alternate initial program load (IPL) of the system. The cartridge identifiers are mounted in the order specified in the CTGORDER parameter.

***NL:** The cartridge is used as a non-labeled tape. The cartridge identifiers are mounted in the order specified in the CTGORDER parameter.

***CNV:** The cartridge identifier is added to the special convenience category. The cartridge identifiers are mounted in the order specified in the CTGORDER parameter.

category-name: Specify the name of a user-defined category. The cartridge identifiers in the category specified are mounted in the order specified in the CTGORDER parameter.

Element 2: Category system

Specifies the name of the system to which the category belongs. The system name is obtained from the pending system name field of a Display Network Attributes (DSPNETA) command. If there is no pending system name, the current system name attribute is used.

Attention:

If a system name is changed, the tape cartridges in media library devices that have the attribute of the system name before it was changed, are no longer valid.

***CURRENT:** The category belongs to the system currently running the command.

system-name: Specify the name of the system to which the category belongs.

Single Values:

***SHARE400:** The cartridge identifier can be shared by all iSeries 400 that are attached to the media library device. The cartridge identifiers are mounted in the order specified in the CTGORDER parameter.

CTGORDER

Specifies the order in which the cartridges are mounted. This parameter is valid only when OPTION(*MOUNTED) is specified.

***SEQ:** The cartridges are mounted in the order they were added or changed to the category specified. The operation ends if the next cartridge in the sequential order is not available.

***NEXTAVAIL:** The cartridges are mounted in a sequential order, but if a cartridge in the order is not available, the next available cartridge is used.

TGTCGY

Specifies the category to which a tape cartridge is changed after it is used. This parameter can be useful when a scratch category is set for use during a save operation. Each category is automatically changed to the specified target category specified after it is used. This parameter is valid only when OPTION(*MOUNTED) is specified.

Element 1: Target category name

***CGY:** The cartridges remain in the category specified on the CGY parameter.

***NOSHARE:** The cartridge identifiers are changed to the *NOSHARE category.

***IPL:** The cartridge identifiers are changed to the *IPL category.

***NL:** The cartridge identifiers are changed to the *NL category.

category-name: Specify the name of a user-defined category. The cartridge identifiers are changed to the specified user-defined category.

Element 2: Target category system

Specifies the name of the system to which the target category belongs. The system name is obtained from the pending system name field of the Display Network Attributes (DSPNETA) command. If there is no pending system name, the current system name attribute is used.

Attention:

If the system name is changed, the tape cartridges in media library devices that have the attribute of the system name before it was changed are no longer valid.

***CURRENT:** The system currently running the command is used.

system-name: Specify the name of the system that the category belongs to.

Single Value:

***SHARE400:** The cartridge identifiers are changed to the *SHARE400 category.

MNTID

Specifies the identifier by which the mounted category session will be known. This parameter is valid only when OPTION(*MOUNTED) or OPTION(*ASSIGN) is specified.

<u>*NONE:</u> This mounted category session is not assigned to any job and will be used by the first job that issues a command to the media library with a volume identifier of *MOUNTED. The mount identifier *NONE cannot be specified with OPTION(*ASSIGN).

mount-identifier: Specify a unique name to identify the mounted category session. A mounted category session known by this mount identifier is created and assigned when the category is mounted. This identifier is then used when assigning the session to another job. The session known by this identifier is deleted when the category is demounted. The mount identifier can then be reused when mounting another category to the media library.

Examples for SETTAPCGY

Example 1: Using mounted category without a mount identifier

```
SETTAPCGY   MLB(LIB01)  OPTION(*MOUNTED)
  CGY(*NOSHARE *CURRENT)
  CTGORDER(*SEQ)  TGTCGY(*IPL)
```

This command sets the tape category to *NOSHARE for a resource in media library device LIB01 on the system currently running this command. Each cartridge that is used is changed to the *IPL category. The order in which the cartridges are used is the exact order in which they were added to or changed in the category. As each cartridge is used and unloaded from the resource, the system automatically chooses and loads the next sequential cartridge from the *NOSHARE category.

Example 2: Using mounted category with a mount identifer

This command sets the tape category to *NOSHARE for a resource in media library device LIB01 on the system currently running this command. The mounted category session is identified by the mount identifier DAILY. Each cartridge that is used is changed to the DAILY1 category. The order in which the cartridges are used is the exact order in which they were added to or changed in the category. As each cartridge is used and unloaded from the resource, the system automatically chooses and loads the next sequential cartridge from the *NOSHARE category.

```
SETTAPCGY MLB(LIB01) OPTION(*RELEASE)
```

This command releases the category session assigned to the job issuing the command. The category is still set to a resource in media library device LIB01 and is available for another job to assign.

```
SETTAPCGY MLB(LIB01) OPTION(*ASSIGN) MNTID(DAILY)
```

This command assigns the mounted category session identified by the mount identifier DAILY to the job issuing the command.

```
SETTAPCGY MLB(LIB01) OPTION(*DEMOUNTED)
```

This command demounts the mounted category from a resource in media library device LIB01. The mount identifier DAILY is now available to use to name another mounted category session.

Error messages for SETTAPCGY

***ESCAPE Messages**

CPF6711
Command not allowed.

CPF6712
Category &4 not mounted.

CPF6713
Category not demounted.

CPF6745
Device &1 not a media library device.

CPF67A6
Category does not exist.

CPF67AD
Category not assigned.

CPF67AE
Category not released.

CPF67E4
Library device function not successful.

CPF9814
Device &1 not found.

CPF9825
Not authorized to device &1.

»

SETUSRBRM (Set User Usage for BRM) Command Description

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

SETUSRBRM Command syntax diagram

Purpose

The Set User Usage for BRM (SETUSRBRM) command allows you to set or reset function usage values for a user or group profile.

This command uses the functional registration facility to set the BRMS recommended usage options for each BRMS function registered with the system. It sets up suage values for using the functions of BRMS, not to control security to BRMS objects such as files in QUSRBRM. You should still use OS/400 security to maintain security of OS/400 objects within BRMS.

Restrictions:

- *SECADM authority is required to use this command.
- The Advanced Functions feature is required to use this command.

Example for SETUSRBRM

Example 1: Setting administrator usage

```
SETUSRBRM
USER(payroll) USAGE(*ADMIN)
```

In this example the group profile 'payroll' is given usage to all BRMS functions.

Error messages for SETUSRBRM

No error messages. «

SIGNOFF (Sign Off) Command Description

SIGNOFF Command syntax diagram

Purpose

The Sign Off (SIGNOFF) command ends an interactive job or causes all group jobs on a work station to end. The user enters this command to sign off at a work station.

Restriction: This command is valid only in an interactive job.

Optional Parameters

LOG Specifies whether the job log for this interactive job is deleted or is included in the job's spooled output for printing. This entry takes precedence over the LOG parameter value specified for the job.

***NOLIST:** The information in the job log is deleted when the job ends.

***LIST:** The job log is spooled for printing along with the rest of the job's spooled output.

DROP Specifies, for switched lines only, whether the switched line attached to the work station is disconnected (dropped) if no other work stations on the same line are signed on. This parameter is ignored if the work station is attached to a nonswitched line.

***DEVD:** The value specified in the DROP parameter of the work station's device description is assumed.

***YES:** The switched line is disconnected when the job is ended, if no other work stations are signed on the line.

***NO:** The switched line is not disconnected when the job is ended.

ENDCNN

Specifies whether to end the connection to the current system. Ending the connection allows the user to bypass the sign-on display of the target system and return to the source system. For communication functions that do not support this option, this parameter is ignored.

***NO:** The connection does not end. The sign-on display of the target system is shown.

***YES:** The connection ends and the user is returned to the source system. No sign-on screen or error messages are shown from the target system.

Examples for SIGNOFF

Example 1: Signing Off and Ending an Interactive Job

```
SIGNOFF
```

This command signs off the user of the work station and ends the interactive job. The switched line is dropped only if specified in the device description of this work station and if no other work station on this line is active. An end-of-job message that gives the job start and stop times is written in the job's log.

Example 2: Printing the Job Log

```
SIGNOFF  LOG(*LIST)  DROP(*NO)
```

This command ends the interactive job, but the switched line is not released. The job log is printed with the job's spooled output.

Example 3: Signing Off and Ending the Connection

```
SIGNOFF  ENDCNN(*YES)
```

This command ends the connection and transfers the user back to the source system.

Error messages for SIGNOFF

None

STREML3270 (Start 3270 Display Emulation) Command Description

STREML3270 Command syntax diagram

Purpose

The Start 3270 Display Emulation (STREML3270) command starts a 3270 device emulation session for a display device. The user can type this command on the command line or from any display station that allows CL commands to be specified. This command can also be issued from a batch job by specifying the display device (DSPDEV) parameter.

The STREML3270 command can be in a CL program specified as the INLPGM for a user profile that is run when the user signs on the display device.

More information on device emulation is in the 3270 Device Emulation Support book.

Required Parameters

EMLCTL

Specifies the name of a binary synchronous communications (BSC) controller description (APPTYPE(*EML)) or Systems Network Architecture (SNA) controller description type of HOST with attached 3270 emulation display devices. When this parameter is specified, the requesting display device is linked to an available 3270 emulation display device on the emulation controller. The controller of the device must be available, and the requester of the command must be authorized to use the controller.

EMLDEV

Specifies the name of a BSC or an SNA emulation display device that is linked to the requesting display device to emulate a specific type of 3270 display device. The requester must be authorized for this device and the device must be available.

EMLLOC

Specifies the emulation remote location name that describes the location of the 3270 emulation display devices. When this parameter is specified, the requesting display device is linked to an available 3270 display device referred to by the remote location. At least one of the emulation devices referred to by the remote location must be available, and the requester of the command must be authorized to use the device. A remote location can refer to as many as 1,016 emulation display devices.

Optional Parameters

DSPDEV

Specifies the name of the display device used for device emulation when the command is in a batch job. The 3270 support tries to acquire the display device by this name; if the display device is acquired, the 3270 device emulation is active on that display device.

***CURRENT:** The current display device name is used for device emulation.

display-device-name: Specify the display device name used for device emulation.

PAGEUP

Specifies an added function for the Page Up (Roll Down) key on the 5250 keyboard when 3270 device emulation is active. This assignment is in effect when the number is not larger than the maximum number of input fields.

***PA2:** The 3270 PA2 function is assigned to the Page Up key.

***NONE:** No 3270 function is assigned to the Page Up key. When there are fewer input fields on the display than allowed by the 5250 display device, the Page Up key has no function.

***F-KEY:** Specify the 3270 function key assigned to the Page Up key.

***PA1-KEY:** Specify the PA1 key assigned to the Page Up key.

***PA3-KEY:** Specify the PA3 key assigned to be the Page Up key.

***CLEAR:** Specify the 3270 Clear function assigned to the Page Up key.

***ERASEINP:** Specify the 3270 Erase Input function key assigned to the Page Up key.

***CSRSLT:** Specify the 3270 Cursor Select Key function key assigned to the Page Up key, and the real Cursor Select Key cannot be used.

PAGEDOWN

Specifies an added function for the Page Down (Roll Up) key on the 5250 keyboard when 3270 device emulation is active. This assignment is in effect when the number is not larger than the maximum number of input fields.

***PA1:** The 3270 PA1 function is assigned to the Page Down key.

***NONE:** No 3270 function is assigned to the Page Down key. When there are fewer input fields on the display than allowed by the 5250 display device, the Page Down key has no function.

***F-KEY:** Specify the 3270 function assigned to the Page Down key.

***PA2-KEY:** Specify the PA2 key assigned to the Page Down key.

***PA3-KEY:** Specify the PA3 key assigned to be the Page Down key.

***CLEAR:** Specify the 3270 Clear function assigned to the Page Down key.

***ERASEINP:** Specify the 3270 Erase Input function assigned to the Page Down key.

***CSRSLT:** Specify the 3270 Cursor Select Key function assigned to the Page Down key, and the real Cursor Select Key cannot be used.

TESTREQ

Specifies an added function for the Test Request key on the 5250 keyboard when a 3270 device emulation is active.

***DFT:** Normal function is assigned to the Test Request key. This is the system default. The normal function is defined in the emulation Help text, and depends on whether the 3270 emulation display device is BSC or SNA. BSC defaults to a 3270 test request function, while SNA defaults to a 3270 system request function.

***CLEAR:** The Clear key is assigned to the Test Request key.

***ERASEINP:** The Erase Input key is assigned to the Test Request key.

CSRSLT

Specifies a function key to emulate the cursor select key.

***NONE:** A physical function key is not assigned to emulate the cursor select key. Instead, the Cursor Select Key defined on the keyboard is used.

***F-KEY:** Specify the function key assigned to emulate the Cursor Select Key, and the Cursor Select Key defined on the keyboard cannot be used.

IGCEMLPC

Specifies whether SNA Japanese 3270 PC emulation or 3270 device emulation is used.

***NO:** SNA Japanese 3270 PC emulation is not used.

***YES:** SNA Japanese 3270 PC emulation is used.

EMLPRTDEV

Specifies the printer device used for SNA Japanese 3270 PC emulation. The printer device is selected after the display device is selected.

***NONE:** SNA Japanese 3270 PC emulation with display emulation is not used. No emulation printer device is selected.

***EMLCTL:** The first available emulation printer device from the specified emulation controller in the EMLCTL parameter is used.

***EMLLOC:** The first available emulation printer device from the specified emulation location in the EMLLOC parameter is used.

emulation-device-description-name: Specify the emulation printer device with the selected emulation display device for SNA Japanese 3270 PC emulation.

INZWAIT

Specifies the first amount of time (in seconds) that the utility waits for the first display from the host system. This parameter only applies to the first read to the host system. If the host system does not send the first display within this time, the emulation session is ended and a message is returned to the requester.

120: The utility waits 120 seconds for the first display from the host system.

***NOMAX:** There is no limit on the amount of time the utility waits for the first display from the host system. This value can be used when the user is not sure when the host system is active to this session. The request can be ended by using the system request and ending request functions.

number-of-seconds: Specify the length of time (in seconds) that the utility waits for the first display from the host system. Valid values range from 1 through 32767.

NUMLCK

Specifies whether the numeric input fields allow only numeric data on a 5250 keyboard.

***EMLDEV:** Numeric shift lock is specified in the emulation device description in the EMLNUMLCK parameter. The DSPDEVD command can be used to display the current EMLNUMLCK setting for the emulation device.

***NO:** Data can be typed in the numeric input fields of a 3270 emulation device.

***YES:** Only numeric data can be typed in the numeric input fields of a 3270 emulation device. Numeric data includes characters ranging from 0 through 9, the symbols + - , . and the blank symbol, which is the character *b* with a slash on the stem.

NULLS

Specifies how beginning and embedded nulls within the 3270 data stream (sent from a 5250 display station) are changed.

***BLANK:** Beginning and embedded nulls are changed to blanks before the 3270 data stream is sent to the host system.

***REMOVE:** Beginning and embedded nulls are removed before the 3270 data stream is sent to the host system.

LOGON

Specifies the sign-on (logon) text that is sent to the host system after starting SNA 3270 device emulation. This text can be used to sign on a specific host application.

This parameter is not allowed if specified for BSC 3270 display emulation, SNA 3270 display station pass-through, or SNA DBCS 3270PC emulation.

***NONE:** No text is sent to the host system after starting 3270 device emulation.

host-logon-command: Specify text that is sent to the host system after starting 3270 device emulation. The text must be enclosed in apostrophes if it contains blanks or other special characters. All apostrophes within the text must be represented by two apostrophes. A maximum of 256 characters can be specified.

WAITRSP

Specifies whether the 3270 emulation device waits until the data received is shown on the workstation display to send a positive response to the host system. The response time recorded by the iSeries 400 may be longer than the time recorded by the host when the emulation device does not wait.

***NO:** The emulation device does not wait to send a positive response. It sends the response as soon as the data received is sent to the workstation display.

***YES:** The emulation device waits until the data received is shown on the workstation display to send a positive response.

ENDCOND

Specifies additional ways in which the SNA 3270 device emulation session can end.

This parameter is not allowed if BSC 3270 display emulation, SNA 3270 display station pass-through, or SNA DBCS 3270PC emulation is specified on the EMLDEV parameter.

***NONE:** No additional ways to end 3270 device emulation are requested.

***DACTLU:** The 3270 display emulation session ends if it receives an SNA DACTLU command from the host system.

There are certain host system applications that issue a DACTLU (deactivate logical unit) before starting, such as Time Sharing Option (TSO), which ends the 3270 display emulation session before the desired application is accessed. This end condition must be avoided when trying to access these applications.

***UNBIND:** The 3270 emulation session ends if it receives an SNA UNBIND from the host system. Consider the following items before selecting this end condition:

- This end condition can be used only when accessing one host application for the duration of the session. An UNBIND occurs while switching from one application to the next, and the 3270 session ends before accessing the second application.
- This end condition can only be used when the communication path to the host system is a simple one. A simple communication path is one that only involves accessing the iSeries 400 on which the STREML3270 command is run, and accessing the host system that contains the desired application. Intermediate systems can exist along this simple path as long as they are not accessed. If intermediate systems are accessed, an UNBIND occurs while switching from one system to the next, and the 3270 display emulation session ends before accessing the desired application.
- There are certain host system applications that issue an UNBIND before starting, such as Time Sharing Option (TSO), which ends the 3270 display emulation session before the desired application is accessed. This end condition must be avoided when trying to access these applications.

ATNEMLMNU

Specifies whether or not the emulation menu is displayed when the Attn key is pressed.

***YES:** The 3270 display emulation menu is displayed when the Attn key is pressed.

***NO:** The 3270 display emulation menu is not displayed when the Attn key is pressed.

FKEYPGM

Specifies a user-exit program and one or more function keys that call the program. When a specified function key is pressed during the 3270 display emulation session and is sent to the host system, the user-exit program is called. When the user-exit program ends, control is returned to the 3270 display emulation session at the point where the function key was pressed.

This parameter is not valid if specified for either BSC 3270 display emulation, SNA 3270 display station pass-through, or SNA DBCS 3270PC emulation.

Note: The user-exit program is called only if the function key is successfully sent to the host system. If the function key fails to be received, an error reset message appears at the bottom of the display suggesting you to try again.

The iSeries 400 user-exit program must be coded to allow for input parameters. The following parameters are passed to the program in the specified order:

1. The function key identifier (10 characters). The identifier of the function key that was pressed. If function key 1 is pressed, the parameter value is *F1. If function key 2 is pressed, the parameter value is *F2, and so on, up to function key 24. The value is left-justified within the parameter.
2. The display name (10 characters). The name of the display on which the 3270 display emulation session is running. The value is left-justified within the parameter.
3. The cursor location (6 characters). The screen location of the cursor at the time the function key was pressed. The first three characters are the row position of the cursor location. The second three characters are the column position of the cursor location. For example, if the cursor location is row 24, column 1 when the function key is pressed, the value of the parameter is 024001. The row and column can be extracted from the variable using substring logic.

Element 1: Program Name

***NONE:** A user-exit program is not associated with any function key.

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 and library of the user-exit program that is called when one of the specified function keys is pressed. The program cannot be a system program.

Element 2: Function Key

***ALLFKEYS:** All function keys call the specified user-exit program.

function-key: Specify a function key to call the user-exit program. A maximum of 24 values can be specified on this parameter.

KBDTYPE

Specifies the type of keyboard language used on the display device.

This parameter does not apply when running SNA 3270 display station pass-through.

***DSPDEV:** The display device description is used.

***SYSVAL:** The value specified in the system value QKBDTYPE is used.

***LCL:** The display device that requested 3270 device emulation is a local display device. The keyboard type is determined from the display device description.

***TRNTBL:** Allows user-defined translate tables to be used. The character translation is defined in the translation tables specified by the TRNTBLOUT and TRNTBLIN parameters. This value is valid for both local and remote display devices.

keyboard-language-identifier: Specify the 3-character identifier for the language group associated with the remote display requesting 3270 emulation. The identifier consists of 3 characters. The languages and associated identifiers are shown below.

Table 1. Keyboard Mapping

Language/Country or Region	Identifier	ASCII Device Group
Albania	ALI	
Arabic X/Basic	CLB	D
Austria/Germany	AGB	A, B
Austria/Germany Euro Currency	AGE	
Austria/Germany Multinational	AGI	A, B
Belgium Multinational	BLI	B
Belgium Multinational Euro Currency	BLM	
Brazilian Portuguese	BRB	
Brazilian Portuguese Euro Currency	BRE	
Bulgaria	BGB	
Canadian French	CAB	A, B
Canadian French Multinational	CAI	A, B
Canadian French Multinational Euro Currency	CAM	
Chinese (Simplified)	RCB	
Chinese (Traditional)	TAB	
Croatia	YGI	
Cyrillic	CYB	
Czech Republic	CSB	
Denmark	DMB	B
Denmark Euro Currency	DME	
Denmark Multinational	DMI	B
Estonia	ESB	
Finland/Sweden	FNB	B
Finland/Sweden Euro Currency	FNE	
Finland/Sweden Multinational	FNI	B
France (Azerty)	FAB	A, B
France (Azerty) Euro Currency	FAE	

Language/Country or Region	Identifier	ASCII Device Group
France (Azerty) Multinational	FAI	A, B
France (Qwerty)	FQB	
France (Qwerty) Multinational	FQI	
Greece [1]	GNB [1]	
Hebrew	NCB	D
Hungary	HNB	
Iceland	ICB	
Iceland Euro Currency	ICE	
Iceland Multinational	ICI	
International	INB	
International Multinational	INI	
Iran (Farsi)	IRB	
Italy	ITB	A, B
Italy Euro Currency	ITE	
Italy Multinational	ITI	A, B
Japan English	JEB	
Japan English Multinational	JEI	
Japan Kanji (for PS/55 and 5295 display stations)	JKB	
Japan Latin Extended	JPB	
Japan United States Basic	JUB	
Japan Katakana (for 5251, 5291, 5292, and 3180 Katakana display stations)	KAB	
Korea	KOB	
Latin-2/ROECE	ROB	
Latvia	LVB	
Lithuania	LTB	
FYR Macedonia (Former Yugoslav Republic)	MKB	
Netherlands	NEB	
Netherlands Euro Currency	NEE	
Netherlands Multinational	NEI	
Norway	NWB	B
Norway Euro Currency	NWE	
Norway Multinational	NWI	B
Poland	PLB	
Portugal	PRB	B
Portugal Euro Currency	PRE	
Portugal Multinational	PRI	B
Romania	RMB	
Russia	RUB	
Serbia (Cyrillic)	SQB	
Serbia (Latin)	YGI	
Slovakia	SKB	
Slovenia	YGI	
Spain	SPB	B
Spain Euro Currency	SPE	
Spain Multinational	SPI	B
Spanish Speaking	SSB	B
Spanish Speaking Multinational Euro Currency	SSE	
Spanish Speaking Multinational	SSI	B
Sweden	SWB	B
Sweden Euro Currency	SWE	
Sweden Multinational	SWI	B
Switzerland/French Multinational	SFI	B
Switzerland/French Multinational Euro Currency	SFM	

Language/Country or Region	Identifier	ASCII Device Group
Switzerland/German Multinational	SGI	B
Switzerland/German Multinational Euro Currency	SGM	
Thai	THB	
Turkey (Qwerty)	TKB	
Turkey (F)	TRB	
Ukraine	UAB	
United Kingdom	UKB	A, B
United Kingdom Euro Currency	UKE	
United Kingdom Multinational	UKI	A, B
United States/Canada	USB	A, B, C
United States/Canada Euro Currency	USE	
United States/Canada Multinational	USI	A, B, C
Urdu	PKB	
Vietnam	VNB	
Languages of the former Yugoslavia	YGI	

Note:

[1] The GNB code is the current identifier for Greece. The GKB code was used prior to V2R1, and continues to be supported, but provides fewer characters than the recommended GNB code.

Note: For example, KBDTYPE(USB) indicates a keyboard using the basic United States/Canada character set.

TRNTBLOUT

Specifies the outgoing translation table used to translate characters sent from the host system to 3270 emulation. This parameter is valid only when KBDTYPE(*TRNTBL) is specified. Both TRNTBLOUT and TRNTBLIN must be specified if KBDTYPE(*TRNTBL) is specified.

***KBDTYPE:** Translation is done using the language specified in the KBDTYPE parameter.

The name of the outgoing translation 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.

user-defined-outgoing-table: Specify the qualified name of the table used for outgoing translation.

TRNTBLIN

Specifies the incoming translation table used to translate characters sent from 3270 emulation to the host system. This parameter is valid only when KBDTYPE(*TRNTBL) is specified. Both TRNTBLOUT and TRNTBLIN must be specified if KBDTYPE(*TRNTBL) is specified.

***KBDTYPE:** Translation is done using the language specified in the KBDTYPE parameter.

The name of the incoming translation 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.

user-defined-incoming-table: Specify the qualified name of the table used for incoming translation.

EMLCFGE

Specifies whether a configuration entry is used for this session. Configuration entries indicate 3270 emulation configuration options. Configuration entries are created with the Add Emulation Configuration Entry (ADDEMLCFGE) command.

***NONE:** No configuration entry is named and the configuration entry defaults are used.

QEMDFTCFGE: The default configuration entry QEMDFTDFGE is used. This entry is shipped with configuration entry defaults, and can be updated with the Change Emulation Configuration Entry (CHGEMLCFGE) command.

configuration-entry-name: Specify the name of the configuration entry to be used. If the configuration entry named does not exist in the configuration file, the configuration entry defaults are used.

Example for STREML3270

```
STREML3270   EMLCTL(TSOHOST)  PAGEUP(*F7)  PAGEDOWN(*F8)
```

This command places the display device into an emulation session that uses the first available device on the controller description TSOHOST for which the user has authority. When there are fewer input fields on the display than the maximum allowed by the 5250 display device and the Page Up key is pressed, an F7 key value is sent to the host system. When the Page Down key is pressed, an F8 key value is sent to the host system.

Error messages for STREML3270

***ESCAPE Messages**

CPF2619

Table &1 not found.

CPF269A

Library parameter is not set to “QSYS ” on call.

CPF269B

T.61 conversion table not found.

CPF85EA

Screen address received from host is larger than screen size.

CPF85EB

3270 device emulation session ended.

CPF85EC

Specifying text on the LOGON parameter is not supported.

CPF85ED

Values other than ENDCOND(*NONE) are not supported.

CPF85E2

3270 display emulation is already active at this job.

CPF85E4
Not authorized to translation table &1 in library &2.

CPF85E5
3270 emulation device &1 is reserved for device &2.

CPF85E6
Translation table &1 in library &2 was not found.

CPF8503
Emulation controller &1 not found.

CPF8504
Controller &1 does not support 3270 emulation.

CPF8505
Emulation device &2 not found.

CPF8506
Emulation location &1 not found.

CPF8507
Display emulation cannot open required file.

CPF8508
Host system did not respond.

CPF851A
Maximum number of shift in and shift out characters exceeded.

CPF8510
Internal error occurred on device &1.

CPF8511
Emulation ended by errors on device &2.

CPF8512
Emulation ended because device &2 was held.

CPF8513
Emulation ended by errors on device &2.

CPF8514
Error recovery stopped on device &1.

CPF8515
3270 emulation session ended by host.

CPF8516
No match between host and device &2.

CPF8517
Received more than maximum number of fields allowed.

CPF8518
Emulation ended because of internal failure in system.

CPF8519
Function check in 3270 emulation.

CPF852A
Values other than FKEYPGM(*NONE) are not supported.

CPF852B
Program &1 not found.

CPF852C
Not authorized to program &1.

CPF8521
Not authorized to controller &1.

CPF8522
Not authorized to emulation device &2.

CPF8523
*NO on the ATNEMLMNU parameter is not supported.

CPF8524
Emulation cannot open its required display file.

CPF8525
KBDTYPE(*LCL) not allowed for remote display devices.

CPF8526
No 3270 display emulation devices available.

CPF8527
Emulation device &2 not available.

CPF8528
Device &2 is not a display emulation device.

CPF8530
Not authorized to use any display emulation device.

CPF8533
Display device not specified in a batch job.

CPF8534
Display device &1 is not available.

CPF8535
Display device &1 not found.

CPF8536
Not authorized to display device &1.

CPF8539
&1 keyboard type not supported by 3270 emulation.

CPF8546
No 3270 display emulation devices available.

CPF8547
No 3270 printer emulation devices available.

CPF8550
Emulation ended due to time-out internal failure.

CPF8551
Emulation ended with error code &1.

CPF8552
Emulation ended because of return code.

CPF8553
BSC controller or device not allowed without translation.

CPF8565
Emulation device &1 not found.

CPF8568
Device &1 not printer emulation device.

CPF8569
Not authorized to use any printer emulation device.

CPF8571
No 3270 printer emulation devices available.

CPF8572
Emulation device &1 not available.

STRARCBRM (Start Archive using BRM) Command Description

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

STRARCBRM Command syntax diagram

Purpose

The Start Archive using BRM (STRARCBRM) command selects a control group and begin the archive process. You can start the archive immediately or you can schedule it using the system job scheduler. Processing can be batch or interactive.

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

Examples for STRARCBRM

Example 1: Producing a Report of Archive Candidates

```
STRARCBRM CTLGRP(ARCH01) SCHTIME(1230)
```

In this example an archive control group ARCH01 will be processed at 12:30 p.m.. A report of all archive candidates is produced.

Example 2: Processing an Archive

```
STRARCBRM CTLGRP(ARCH01) OPTION(*ARCHIVE)
SCHTIME(1230)
```

In this example the archive control group ARCH01 will be archived at 12:30 p.m..

Error messages for STRARCBRM

None

STRASPBAL (Start ASP Balance) Command Description

STRASPBAL Command syntax diagram

Purpose

The Start ASP Balance (STRASPBAL) command allows the user to start the auxiliary storage pool (ASP) balancing function for one or more auxiliary storage pools (ASPs). » Four types of ASP balancing can be started: «

1. **Capacity balancing** - The data on all the units within the ASP will be balanced so each unit has an equal percentage of used and unused space. This is useful when new units are added to an ASP. Instead of having several units with the majority of the data and several new ones with no data, the user can spread the data evenly across all the units.
2. **Usage balancing** - The high-use and low-use data on each unit in the ASP is redistributed to balance the arm utilization of each unit within the specified ASP. *USAGE balancing cannot be done until the trace controlled by the Trace ASP Balance (TRCASPBAL) command has collected usage statistics. TRCASPBAL starts the trace function that will identify the high-use and low-use data on each unit. This information will be used when the Start ASP Balance (STRASPBAL) TYPE(*USAGE) command is run. After this balance activity has run to completion the trace information will be cleared. Usage balancing is useful when the ASP contains large capacity disk units.
3. **Hierarchical Storage Management balancing** - The high-use and low-use data on each unit in the ASP is redistributed so that the high-use data resides on high performance units and the low-use data resides on low performance units. The ASP being balanced in this manner must contain the correct mixture of slow and fast units to perform this type of balance. This allows the low-use data on the high performance units to be moved to low performance units. Data that is high-use that resides on slow units will be moved to the fast disk units. *HSM balancing cannot be done until the trace controlled by the Trace ASP Balance (TRCASPBAL) command has collected usage statistics. TRCASPBAL starts the trace function that will identify the high-use and low-use data on each unit. This information will be used when the Start ASP Balance (STRASPBAL) TYPE(*HSM) command is run. After this balance activity has run to completion the trace information will be cleared. *HSM balancing is useful when the ASP contains compressed disk units.
4. » **Move data from units** - This option can be used to reduce the down time associated with removing a disk unit. A unit that is scheduled for removal can be marked to end allocations by specifying UNIT(*unit-number*) and TYPE(*ENDALC). This will keep new allocations away from this unit. For all units marked *ENDALC, specifying TYPE(*MOVDTA) will move data from the marked units to other units in the same ASP. To resume allocations for units marked *ENDALC, specify UNIT(*unit-number*) and TYPE(*RSMALC). New allocations will once again be allowed to this unit. The Check ASP Balance (CHKASPBAL) command can be used to determine which units are currently marked *ENDALC.«

The user may specify a time limit that the function is to run for each ASP being balanced or the balance can be set to run to completion. If the balance function needs to be ended, use the End ASP Balance (ENDASPBAL) command. A message will be sent to the system history (QHST) log when the balancing function is started for each ASP. A message will also be sent to the QHST log when the balancing function completes or is ended.

If the balance function is run for a few hours and then stopped, it will continue from where it left off when the balance function restarts. This allows the balancing to be run during off hours over a several day period.

For more information about ASP balancing see the Hierarchical Storage Management book.

Restrictions: You must have *ALLOBJ special authority to use this command. »

Required Parameter

TYPE Specifies the type of auxiliary storage pool (ASP) balancing that is to be done.

***CAPACITY:** The capacity balance function will be started for the specified ASP (ASP or ASPDEV parameter).

***USAGE:** The usage balance function will be started for the specified ASP (ASP or ASPDEV parameter).

***HSM:** The hierarchical storage management balance function will be started for the specified ASP (ASP or ASPDEV parameter).

»

***MOVDTA:** Data will be moved off all units marked *ENDALC to other units in the same ASP. The CHKASPBAL command can be used to determine which units are marked *ENDALC.

***ENDALC:** New allocations will no longer go to the specified units (UNIT parameter). However, the system will use these units for new allocations to avoid 'Out of storage' conditions.

***RSMALC:** Resume allocations to the specified units (UNIT parameter). If the unit has a much lower percentage used than other units in the ASP, run STRASPBAL again specifying TYPE(*CAPACITY) for the ASP that contains the unit. This will keep new allocations balanced across the units in the ASP, instead of going mostly to this unit.

Optional Parameters

ASP Specifies the auxiliary storage pool (ASP) for which the ASP balancing function is to be started.

***ALL:** ASP balancing will be started for the system ASP (ASP number 1) and all basic ASPs (ASP numbers 2-32) defined to the system.

auxiliary-storage-pool-number: Specify the ASP for which ASP balancing is to be started. Valid ASP numbers are 1 to 32. Up to 32 ASP numbers may be specified.

ASPDEV
Specifies the name of the auxiliary storage pool (ASP) device for which the ASP balancing is to be started.

***ALLAVL:** ASP balancing will be started for all ASP devices that currently have a status of 'Available'.

auxiliary-storage-device-name: Specify the name of the independent ASP device for which ASP balancing is to be started. Up to 32 ASP device names may be specified.

UNIT Specifies the unit number for which new allocations are to be ended (*ENDALC) or for which new allocations are to resume (*RSMALC). Specifying TYPE(*MOVDTA) will start moving data off the units marked *ENDALC.

unit-number: Specify the unit number for which new allocations are to end (*ENDALC) or resume (*RSMALC). Up to 300 unit numbers may be specified. «

TIMLMT
Specifies the amount of time, in minutes, that the ASP balancing function will be allowed to run. When the time limit is reached the function will end. For example, if ASP(*ALL) is specified and the machine has four ASPs configured and TIMLMT(60) is specified, four balance functions are started and each can run 60 minutes. If the balancing of any ASP has not completed after 60 minutes, it will be forced to end. This allows you to do ASP balancing incrementally. The balance function will not run across an IPL.

***NOMAX:** There is no time limit for the balance function. For a large ASP the balance function can take a long time to complete. If you start the balance function with *NOMAX and you want to force the function to end, you can use the End ASP Balance (ENDASPBAL) command.

time-limit: Specify the time limit that the balance function will be allowed to run. Valid values range from 1 to 9999 minutes.

Examples for STRASPBAL

Example 1: Starting ASP Balancing for ASP 1

```
STRASPBAL  ASP(1) TIMLMT(*NOMAX) TYPE(*CAPACITY)
```

»

This command starts the *CAPACITY ASP balance function for ASP 1. The balance function will run until the capacity for each of the units has been balanced.

Example 2: Starting ASP Balancing with a Time Limit

```
STRASPBAL  ASP(*ALL) TIMLMT(60) TYPE(*USAGE)
```

This command starts the *USAGE ASP balance function for the system ASP and each configured basic ASP for which the TRCASPBAL command has been run. Each balance function will have a time limit of sixty minutes. After sixty minutes, any balance functions which have not completed will be ended.

Example 3: Starting ASP Balancing for an ASP Device

```
STRASPBAL  ASPDEV(MYASP1) TIMLMT(*NOMAX) TYPE(*CAPACITY)
```

This command starts the *CAPACITY ASP balance function for the ASP device named MYASP1. The balance function will run until complete.

Example 4: Preparing to Remove Units

```
STRASPBAL  UNIT(11 12 13) TYPE(*ENDALC)
STRASPBAL  TYPE(*MOVDTA)
```

The first STRASPBAL command marks units 11, 12 and 13 to no longer receive new allocations. The second STRASPBAL command begins to move data off the marked units. It is recommended that the *MOVDTA ASP balancing function be done during an off-peak time.

Error messages for STRASPBAL

***ESCAPE Messages**

CPF18AB
ASP balancing for ASP &2 already started.

CPF18AD
ASP &2 must contain more than a single unit.

CPF18AE
ASP &2 does not contain trace data.

CPF18AF
ASP &2 does not contain mixed unit types.

CPF18B1
Trace function currently running for ASP &2.

CPF18B3
Balance type not valid for ASP &2.

«

STRBKUBRM (Start Backup Using BRM) Command Description

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

STRBKUBRM Command syntax diagram

Purpose

The Start Backup using BRM (STRBKUBRM) command selects a control group to back up. You can start the backup immediately or you can schedule it using the system job scheduler. Processing can be batch or interactive.

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

Example for STRBKUBRM

Example 1: Backing up a Control Group

```
STRBKUBRM CTLGRP(BACK01)
```

In this example a backup control group BACK01 will be submitted to batch and processed immediately.

Error messages for STRBKUBRM

None

STRCLNUP (Start Cleanup) Command Description

STRCLNUP Command syntax diagram

Purpose

The Start Cleanup (STRCLNUP) command checks to determine whether the cleanup operation is allowed. The cleanup operation is allowed if ALWCLNUP(*YES) is specified on the Change Cleanup (CHGCLNUP) command.

A batch job that controls the cleanup operation and the power on/off schedule is submitted to the job queue specified on the CHGCLNUP command. This cleanup control job submits individual batch jobs to the same job queue each day at the time specified on the STRTIME parameter of the CHGCLNUP command. These batch jobs do the actual cleanup of the objects specified on the CHGCLNUP command if cleanup is allowed.

More information is in the CHGCLNUP (Change Cleanup) command description and the Basic System Operations topic in the Information Center.

Restriction: The user of this command must have *JOBCTL authority and must be authorized to the QPGMR user profile.

Optional Parameter

OPTION

Specifies when the cleanup operation is started.

***SCHED:** The cleanup operation is started as specified on the STRTIME parameter of the CHGCLNUP command or on the Change Cleanup Options display.

***IMMED:** The cleanup operation is started immediately.

Examples for STRCLNUP

Example 1: Starting Cleanup Operation as Scheduled

```
STRCLNUP
```

This command starts the cleanup operation as specified.

Example 2: Starting Cleanup Operation as Specified on CHGLNUP Command

```
STRCLNUP  OPTION(*SCHED)
```

This command starts the cleanup operation as specified on the STRTIME parameter of the CHGCLNUP command or on the Change Cleanup Options display.

Example 3: Starting Cleanup Operation Immediately

```
STRCLNUP  OPTION(*IMMED)
```

This command starts the cleanup operation immediately.

Error messages for STRCLNUP

***ESCAPE Messages**

CPF1E2A
Unexpected error in QSYSSCD job.

CPF1E2B
Power scheduler and cleanup options not found.

CPF1E3A
Not authorized to start cleanup.

CPF1E3C
Job queue &2/&1 not found.

CPF1E3D
Library &1 for JOBQ parameter not found.

CPF1E33
Cleanup options or power schedule in use by another user.

CPF1E34
Error occurred starting &1 job.

CPF1E36
Cleanup has not been started.

CPF1E37
Cleanup has already been started or is scheduled to run.

CPF1E38
Cleanup not allowed to run.

CPF1E39
Not authorized to job queue used for cleanup.

CPF1E99
Unexpected error occurred.

»

STRCLUNOD (Start Cluster Node) Command Description

STRCLUNOD Command syntax diagram

Purpose

The Start Cluster Node (STRCLUNOD) command is used to start Cluster Resource Services on a node in the cluster. If Cluster Resource Services is successfully started on the node specified, the status of the node will be set to Active. Cluster Resource Service jobs (QCSTCTL, QCSTCRGM, and a job for each cluster resource group object in the cluster) will be started in the QSYSWRK subsystem on the node just started.

As part of starting the node, Cluster Resource Services checks the allow add to cluster (ALWADDCLU) network attribute to see whether the node being started should be part of the cluster and whether to validate the cluster request through the use of X.509 digital certificates. If validation is required, the requesting node and the node being added must have the following installed on the systems:

- OS/400 option 34 (Digital Certificate Manager)
- Cryptographic Access Provider Product (AC2 or AC3)

If the cluster is partitioned, this command may be used to start nodes in the partition running this command.

The potential node version of the node being started must be equal to the current cluster version or up to one level higher than the current cluster version. The potential node version and the current cluster version can be retrieved by using the Display Cluster Information (DSPCLUINF) command.

If the node being started is in a device domain, then OS/400 option 41, HA Switchable Resources, must be installed and a valid license key must exist on that node.

Restrictions

1. To use this command you must have *IOSYSCFG authority.
2. This command cannot be called from a cluster resource group exit program.
3. The node being started must exist in the cluster membership list.
4. If all nodes have a status of New, this command must be invoked on the node on which the cluster was originally created.
5. The node to be started must be IP reachable (TCP/IP is active and the INETD server is started).
6. The first time a node is started, this command must be invoked on a node that is active.

Required Parameters

CLUSTER
Specifies the name of the cluster that contain the node being started.

cluster-name: Specify the name of the cluster.

NODE Specifies the node identifier being started.

node-identifier: Specify the name of the node being started.

Example for STRCLUNOD

```
STRCLUNOD CLUSTER(MYCLUSTER) NODE(NODE01)
```

This command starts Cluster Resource Services on node NODE01 for cluster MYCLUSTER.

Error messages for STRCLUNOD

***ESCAPE Messages**

CPF0001
Error found on &1 command.

«

Printed in U.S.A.