IBM

IBM Systems - iSeries

Backup and Recovery APIs

Version 5 Release 4

IBM Systems - iSeries
Backup and Recovery APIs

Version 5 Release 4

> **Note**
>
> Before using this information and the product it supports, be sure to read the information in
> "Notices," on page 235.

Contents

Backup and Recovery APIs

The Backup and Recovery APIs list the contents of a save file and save a list of objects.

For information about planning a backup and recovery strategy, see the Backup and recovery topic. For information about procedures for saving and restoring information on your system, see the Backup and recovery topic, as well as the Backup and Recovery ![book icon] book.

The Backup and Recovery APIs are:

- "Change Backup Schedule (QEZCHBKS) API" on page 2 (QEZCHBKS) allows the user to change the Operational Assistant backup schedules.
- "Change Job Media Library Attributes (QTACJMA) API" on page 8 (QTACJMA) API changes the specified job's settings for the media library attributes.
- "Change Object Backup List (QEZCHBKL) API" on page 11 (QEZCHBKL) changes the backup type for a list of objects that are specified by the user.
- "Create Media Definition (QSRCRTMD, QsrCreateMediaDefinition) API" on page 14 (QSRCRTMD, QsrCreateMediaDefinition) creates a media definition specified by the user.
- "Delete Media Definition (QSRDLTMD, QsrDeleteMediaDefinition) API" on page 20 (QSRDLTMD, QsrDeleteMediaDefinition) deletes a media definition specified by the user.
- "Dump Device (QTADMPDV) API" on page 22 (QTADMPDV) collects information for your IBM service representative for use immediately after a suspected device and/or tape management system failure.
- "Free Object (QTAFROBJ) API" on page 25 (QTAFROBJ) 'suspends' a document object specified by the caller of the API.
- "List Save File (QSRLSAVF) API" on page 27 (QSRLSAVF) lists the contents of a save file.
- "Open List of Objects to be Backed Up (QEZOLBKL) API" on page 35 (QEZOLBKL) retrieves an open list of the objects that are to be backed up.
- "Restore from Application (QaneRsta) API" on page 41 (QaneRsta) enables an application to provide the restore records that are required for a restore-from-save-file operation.
- "Restore Object (QsrRestore) API" on page 46 (QsrRestore) restores a copy of one or more objects that can be used in the integrated file system.
- ≫ "Restore Object List (QSRRSTO) API" on page 60 (QSRRSTO) restores a list of objects or spooled files specified by the user. ≪
- "Retrieve Backup Detail (QEZRTBKD) API" on page 84 (QEZRTBKD) retrieves more detailed information about the library or folder that is to be backed up.
- "Retrieve Backup History (QEZRTBKH) API" on page 86 (QEZRTBKH) retrieves information about the backup status and history into a single variable in the calling program.
- "Retrieve Backup Options (QEZRTBKO) API" on page 94 (QEZRTBKO) returns the backup options for the requested backup type.
- "Retrieve Backup Schedule (QEZRTBKS) API" on page 99 (QEZRTBKS) returns information about when the Operational Assistant backups are scheduled to be run.
- "Retrieve Cartridge Filter (QTARCTGF) API" on page 103 (QTARCTGF) retrieves the currently defined cartridge filter for the system.
- "Retrieve Cartridge Information (QTARCTGI) API" on page 105 (QTARCTGI) retrieves a list of the cartridges in a tape library device and their attributes.
- "Retrieve Category List (QTARCGYL) API" on page 111 (QTARCGYL) retrieves a list of the categories currently defined on the system.

- "Retrieve Device Capabilities (QTARDCAP) API" on page 113 (QTARDCAP) retrieves information that is associated with a specified tape device description or tape resource name.
- "Retrieve Device Information (QTARDINF) API" on page 125 (QTARDINF) retrieves information that is associated with a specified device description.
- "Retrieve Device Status (QTARDSTS) API" on page 127 (QTARDSTS) retrieves dynamic status information for the specified device and for any currently mounted tape cartridge.
- "Retrieve Job Media Library Attributes (QTARJMA) API" on page 135 (QTARJMA) retrieves the specified job's current settings for the media library attributes.
- "Retrieve Media Definition (QSRRTVMD, QsrRetrieveMediaDefinition) API" on page 138 (QSRRTVMD, QsrRetrieveMediaDefinition) retrieves a media definition specified by the user.
- "Retrieve Tape Labels (QTARTLBL) API" on page 144 (QTARTLBL) retrieves the label information for the files on a tape cartridge.
- "Save Object (QsrSave) API" on page 150 (QsrSave) saves a copy of one or more objects that can be used in the integrated file system.
- "Save Object List (QSRSAVO) API" on page 166 (QSRSAVO) saves a list of objects specified by the user.
- "Qp0lSaveStgFree()—Save Storage Free" on page 192 (Qp0lSaveStgFree()) calls a user-supplied exit program to save an *STMF iSeries object type and, upon successful completion of the exit program, frees the storage for the object and marks the object as storage freed.
- "QlgSaveStgFree()—Save Storage Free (using NLS-enabled path name)" on page 196 (QlgSaveStgFree()) calls a user-supplied exit program to save an *STMF iSeries object type and, upon successful completion of the exit program, frees the storage for the object and marks the object as storage freed.
- "Save to Application (QaneSava) API" on page 196 (QaneSava) enables an application to receive the save records that are generated by a save-to-save-file operation.
- "Set Cartridge Filter (QTASCTGF) API" on page 202 (QTASCTGF) sets a filter that defines the cartridges that can be used by tape library devices on the system.

The Backup and Recovery exit programs are:
- "Restore from Application Exit Program" on page 205 enables an application program to provide the restore records that are required for a restore-from-save-file operation using the Restore from Application (QaneRsta) API.
- "Save Storage Free Exit Program" on page 207 is called by the "Qp0lSaveStgFree()—Save Storage Free" on page 192 API to save an *STMF iSeries object type.
- "Save to Application Exit Program" on page 208 enables an application program to receive the save records that are generated by a save-to-save-file operation using the Save to Application (QaneSava) API.
- "Storage Extension Exit Program" on page 211 provides the capability to use storage extension.
- "Tape Management Exit Program" on page 215 provides the function to monitor and control the use of volumes and devices used by the operating system for most tape operations.

Top | APIs by category

APIs

These are the APIs for this category.

Change Backup Schedule (QEZCHBKS) API

Required Parameter Group:

| 1 | Input structure | | Input | Char(*) |

2	Length of input structure	Input	Binary(4)
3	Format name	Input	Char(8)
4	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Change Backup Schedule (QEZCHBKS) API allows the user to change the Operational Assistant backup schedules.

Authorities and Locks

Special Authority
      *JOBCTL and *SAVSYS

User Index Authority
      *CHANGE

User Index Lock
      *EXCL

Required Parameter Group

Input structure
      INPUT; CHAR(*)

      The variable that contains the backup schedule changes. The layout of this parameter is defined by the format name parameter.

Length of input structure
      INPUT; BINARY(4)

      Length of the change request structure. A minimum length of 58 is required for the CBKS0100 format.

Format name
      INPUT; CHAR(8)

      The format of the input structure data. Format CBKS0100 contains the information regarding changes to the Operational Assistant backup schedule. For more information, see "CBKS0100 Format."

Error code
      I/O; CHAR(*)

      The structure in which to return error information. For the format of the structure, see Error Code Parameter.

CBKS0100 Format

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Hours before backup to send load-tape message
4	4	BINARY(4)	Occurrence of week in month to run backup
8	8	CHAR(1)	Run backup using this schedule
9	9	CHAR(1)	Sunday backup
10	A	CHAR(6)	Sunday backup time
16	10	CHAR(1)	Monday backup

Offset			
Dec	Hex	Type	Field
17	11	CHAR(6)	Monday backup time
23	17	CHAR(1)	Tuesday backup
24	18	CHAR(6)	Tuesday backup time
30	1D	CHAR(1)	Wednesday backup
31	1F	CHAR(6)	Wednesday backup time
37	22	CHAR(1)	Thursday backup
38	23	CHAR(6)	Thursday backup time
44	2C	CHAR(1)	Friday backup
45	2D	CHAR(6)	Friday backup time
51	33	CHAR(1)	Saturday backup
52	34	CHAR(6)	Saturday backup time

Field Descriptions

Friday backup. The backup type to be performed on Friday. Possible values follow:

1	*DAILY
2	*WEEKLY
3	*MONTHLY
4	*WEEKMONTH. The weekly backup options are used to run the backup every week except for the week that the monthly backup is to occur. The monthly backup week is determined by the value that the user specifies for the occurrence of week in month to run backup field.
9	*SAME. No change is made to the current backup schedule for the specified day of the week.
blank	No backup is scheduled for the specified day of the week.

Friday backup time. The time that the backup should occur on Friday. Possible values follow:

HHMMSS	The time that the backup operation should occur for the specified day of the week. A 24-hour format is used.
blank	No backup operations are scheduled to be performed for the specified day of the week.
*SAME	No change should be made to the current backup operations that are scheduled for the specified day of the week.

Hours before backup to send load-tape message. The number of hours prior to a backup for a system-operator load-tape-message reminder to be sent. The possible values follow:

0	*NOMSG. No message is sent.
1-24	The number of hours prior to backup to send the message.
-1	*SAME. No change is made to the scheduled hours before backup to send the load-tape message.

Monday backup. The backup type to be performed on Monday. Possible values follow:

1	*DAILY
2	*WEEKLY
3	*MONTHLY

4	*WEEKMONTH. The weekly backup options are used to run the backup every week except for the week that the monthly backup is to occur. The monthly backup week is determined by the value that the user specifies for the occurrence of week in month to run backup field.
9	*SAME. No change is made to the current backup schedule for the specified day of the week.
blank	No backup is scheduled for the specified day of the week.

Monday backup time. The time that the backup should occur on Monday. Possible values follow:

HHMMSS	The time that the backup operation should occur for the specified day of the week. A 24-hour format is used.
blank	No backup operations are scheduled to be performed for the specified day of the week.
*SAME	No change should be made to the current backup operations that are scheduled for the specified day of the week.

Occurrence of week in month to run backup. The week of the month that you want the backup to occur when the backup type is *MONTHLY or *WEEKMONTH. Possible values follow:

-1	*SAME. No changes are made to this value.
0	No monthly backups are scheduled. (If there are no days specified with *MONTHLY or *WEEKMONTH, this value is not used and is ignored.)
1-4	The corresponding week of the month during which the monthly backup occurs.
5	*LAST. The monthly backup should be run on the last week for any given month.

Run backup using this schedule. Whether the backup schedule should be used to run backups. Possible values follow:

0	No. Save all the schedule values, but do not run the backups.
1	Yes. Allow backups to run according to this schedule.
blank	*SAME. Use the existing Run backup using this schedule value.

Saturday backup. The backup type to be performed on Saturday. Possible values follow:

1	*DAILY
2	*WEEKLY
3	*MONTHLY
4	*WEEKMONTH. The weekly backup options that are used to run the backup every week except for the week that the monthly backup is to occur. The monthly backup week is determined by the value that the user specifies for the occurrence of week in month to run monthly backup field.
9	*SAME. No change is made to the current backup schedule for the specified day of the week.
blank	No backup is scheduled for the specified day of the week.

Saturday backup time. The time the backup should occur on Saturday. Possible values follow:

HHMMSS	The time that the backup operation should occur for the specified day of the week. A 24-hour format is used.
blank	No backup operations are scheduled to be performed for the specified day of the week.
*SAME	No change should be made to the current backup operations that are scheduled for the specified day of the week.

Sunday backup. The backup type to be performed on Sunday. Possible values follow:

1	*DAILY
2	*WEEKLY

3	*MONTHLY
4	*WEEKMONTH. The weekly backup options are used to run the backup every week except for the week that the monthly backup is to occur. The monthly backup week is determined by the value that the user specifies for the occurrence of week in month to run backup field.
9	*SAME. No change is made to the current backup schedule for the specified day of the week.
blank	No backup is scheduled for this day of the week.

Sunday backup time. The time that the backup should occur on Sunday. Possible values follow:

HHMMSS	The time that the backup operation should occur for the specified day of the week. A 24-hour format is used.
blank	No backup operations are scheduled to be performed for the specified day of the week.
*SAME	No change should be made to the current backup operations that are scheduled for the specified day of the week.

Thursday backup. The backup type to be performed on Thursday. Possible values follow:

1	*DAILY
2	*WEEKLY
3	*MONTHLY
4	*WEEKMONTH. The weekly backup options are used to run the backup every week except for the week that the monthly backup is to occur. The monthly backup week is determined by the value that the user specifies for the occurrence of week in month to run backup field.
9	*SAME. No change is made to the current backup schedule for the specified day of the week.
blank	No backup is scheduled for the specified day of the week.

Thursday backup time. The time the backup should occur on Thursday. Possible values follow:

HHMMSS	The time that the backup operation should occur for the specified day of the week. A 24-hour format is used.
blank	No backup operations are scheduled to be performed for the specified day of the week.
*SAME	No change should be made to the current backup operations that are scheduled for the specified day of the week.

Tuesday backup. The backup type to be performed on Tuesday. Possible values follow:

1	*DAILY
2	*WEEKLY
3	*MONTHLY
4	*WEEKMONTH. The weekly backup options are used to run the backup every week except for the week that the monthly backup is to occur. The monthly backup week is determined by the value that the user specifies for the occurrence of week in month to run backup field.
9	*SAME. No change is made to the current backup schedule for the specified day of the week.
blank	No backup is scheduled for the specified day of the week.

Tuesday backup time. The time that the backup should occur on Tuesday. Possible values follow:

HHMMSS	The time that the backup operation should occur for the specified day of the week. A 24-hour format is used.
blank	No backup operations are scheduled to be performed for the specified day of the week.
*SAME	No change should be made to the current backup operations that are scheduled for the specified day of the week.

Wednesday backup. The backup type to be performed on Wednesday. Possible values follow:

1	*DAILY
2	*WEEKLY
3	*MONTHLY
4	*WEEKMONTH. The weekly backup options are used to run the backup every week except for the week that the monthly backup is to occur. The monthly backup week is determined by the value that the user specifies for the occurrence of week in month to run backup field.
9	*SAME. No change is made to the current backup schedule for the specified day of the week.
blank	No backup is scheduled for the specified day of the week.

Wednesday backup time. The time the backup should occur on Wednesday. Possible values follow:

HHMMSS	The time that the backup operation should occur for the specified day of the week. A 24-hour format is used.
blank	No backup operations are scheduled to be performed for the specified day of the week.
*SAME	No change should be made to the current backup operations that are scheduled for the specified day of the week.

Error Messages

Message ID	Error Message Text
CPF1061 E	Time not valid.
CPF1099 E	Subsystem not started because system ending.
CPF1629 E	Not authorized to job schedule &1.
CPF1637 E	Job schedule &1 in library &2 in use.
CPF1EC0 E	Job schedule in use by another user.
CPF1EC3 E	Not authorized to Backup Schedule.
CPF1EC4 E	Cannot display Backup Schedule.
CPF1EC5 E	Backup option &1 is not valid.
CPF1EC6 E	Value &1 for run backup not valid.
CPF1EC8 E	Value &1 for hours before backup not valid.
CPF1EC9 E	Value &1 for occurrence of week not valid.
CPF1E99 E	Unexpected error occurred.
» CPF1EE3 E	Not authorized to backup options. «
CPF24B4 E	Severe error while addressing parameter list.
CPF3C17 E	Error occurred with input data parameter.
CPF3C21 E	Format name &1 is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF8122 E	&8 damage on library &4.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9807 E	One or more libraries in library list deleted.
CPF9808 E	Cannot allocate one or more libraries on library list.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9830 E	Cannot assign library &1.
CPF9838 E	User profile storage limit exceeded.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V3R7

Change Job Media Library Attributes (QTACJMA) API

Required Parameter Group:

1	Media library attributes description	Input	Char(*)
2	Length of media library attributes description	Input	Binary(4)
3	Format name	Input	Char(8)
4	Qualified job name	Input	Char(26)
5	Internal job identifier	Input	Char(16)
6	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

The Change Job Media Library Attributes (QTACJMA) API changes the specified job's settings for the media library attributes. For more information on tape management, see Tape information in the Storage Solutions topic.

Authorities and Locks

≫ Device Description Authority
        *CHANGE ≪

Job Authority
        *JOBCTL, if the job for which information is changed has a different user profile from that of the job that calls the QTACJMA API. *JOBCTL special authority is required when changing or replacing the resource allocation priority.

Required Parameter Group

Media library attributes description
        INPUT; CHAR(*)

        The media library attributes. Either the entire list of attributes will be replaced or only specified entries will be changed by this specification.

Length of media library attributes description
        INPUT; BINARY(4)

        The length of the media library attributes description, in bytes.

Format name
        INPUT; CHAR(8)

        The format name CJMA0100 is the only valid format name used by this API. For more information, see "CJMA0100 Format" on page 9.

Qualified job name
        INPUT; CHAR(26)

        The name of the job for which information is to be changed. The qualified job name has three parts:

Job name
        CHAR(10). A specific job name or the following special value:

		The job that this program is running in. The rest of the qualified job name parameter must be blank.
*INT		The internal job identifier locates the job. The user name and job number must be blank.

User name
>   CHAR(10). A specific user profile name, or blanks when the job name is a special value or *INT.

Job number
>   CHAR(6). A specific job number, or blanks when the job name specified is a special value or *INT.

Internal job identifier
>   INPUT; CHAR(16)

>   The internal identifier for the job. The List Job (QUSLJOB) API creates this identifier. If you do not specify *INT for the job name parameter, this parameter must contain blanks. With this parameter, the system can locate the job more quickly than with a job name.

Error code
>   I/O; CHAR(*)

>   The structure in which to return error information. For the format of the structure, see Error Code Parameter.

CJMA0100 Format

The following table lists the fields for the media library attributes description in the CJMA0100 format. For more information about each field, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	CHAR(10)	Option
10	A	CHAR(2)	Reserved
12	C	BINARY(4)	Number of device entries
Offsets vary. These fields repeat in the order listed, for each media library device that is to have attributes defined.		CHAR(10)	Media library device
		CHAR(6)	Reserved
		BINARY(4)	Resource allocation priority
		BINARY(4)	Wait time for initial amount
		BINARY(4)	Wait time for end of volume mount
		CHAR(4)	Reserved

Field Descriptions

Media library device. The name of the media library device that the attributes apply to. The special values supported are:

*ALL	The attributes apply to all media libraries. The value *ALL is only allowed when changing the attributes and must be the first and only device entry.

DEFAULT   The attributes apply to all media libraries that do not have specific attributes defined for the specified job. The *DEFAULT device is only allowed when replacing the attribute list and must be specified as the first device entry.

Number of device entries. The number of entries in the device list changed for this format. There must be at least one entry defined. The maximum number of device entries allowed is 1000.

Option. An option specifying the action to take. Special values are:

CHANGE   The media library attributes are changed by using the device entries specified in the media library attributes description. If an entry already exists for a specified device, that entry will be replaced. If no entry exists for a specified device, an entry will be created.

REPLACE   The entire list of media library attributes are replaced by the device entries specified in the media library attributes description. The first entry must be for the *DEFAULT device.

Reserved. This field must be set to hexadecimal zeros.

Resource allocation priority. The priority the specified job will be given when the job requests a tape resource within a media library device.

Valid values range from 1 (highest) through 99 (lowest).

Exceptions:
- Value of -1 implies *SAME. The resource allocation priority will remain the same. This value is only allowed for the *CHANGE option.
- Value of -2 implies *DEV. The priority specified in the device description will be used when the job requests a tape resource.
- Value of -31 implies *JOB. The specified job's run-time priority will be used for the resource allocation priority when the job requests a tape resource.

Wait time for end of volume mount. The maximum amount of time, in minutes, a request will wait for the allocation of a tape resource to mount the next volume after the end of volume is reached. Valid values range from 1 through 600.

Exceptions:
- Value of -1 implies *SAME. The wait time for the end of volume mount will remain the same. This value is only allowed for the *CHANGE option.
- Value of -2 implies *DEV. The end of volume mount wait time specified in the device description will be used.
- Value of -8 implies *NOMAX. The specified job will wait until a resource becomes available.
- Value of -31 implies *JOB. The specified job's default wait time will be used to calculate the wait time. The time is calculated by rounding the default wait time, in seconds, to the next highest minute.
- Value of -32 implies *IMMED. The specified job will not wait for a resource to become available.

Wait time for initial mount. The maximum amount of time, in minutes, a request will wait for the allocation of a tape resource to mount the first volume. Valid values range from 1 through 600.

Exceptions:
- Value of -1 implies *SAME. The wait time for the initial mount will remain the same. This value is only allowed for the *CHANGE option.
- Value of -2 implies *DEV. The initial mount wait time specified in the device description will be used.
- Value of -8 implies *NOMAX. The specified job will wait until a resource becomes available.

- Value of -31 implies *JOB. The specified job's default wait time will be used to calculate the wait time. The time is calculated by rounding the default wait time, in seconds, to the next highest minute.
- Value of -32 implies *IMMED. The specified job will not wait for a resource to become available.

Error Messages

Message ID	Error Message Text
CPF1343 E	Job &3/&2/&1 not valid job type for function.
CPF136A E	Job &3/&2/&1 not active.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3C51 E	Internal job identifier not valid.
CPF3C52 E	Internal job identifier no longer valid.
CPF3C53 E	Job &3/&2/&1 not found.
CPF3C54 E	Job &3/&2/&1 currently not available.
CPF3C55 E	Job &3/&2/&1 does not exist.
CPF3C58 E	Job name specified is not valid.
CPF3C59 E	Internal identifier is not blanks and job name is not *INT.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF6708 E	Command ended due to error.
CPF67B1 E	Option value &1 not valid.
CPF67B2 E	Number of devices entries &1 not valid.
CPF67B3 E	Media library device &1 not valid.
CPF67B4 E	Value &1 in field &2 not valid.
CPF67B5 E	&3/&2/&1 not authorized to change attribute.
CPF67B6 E	&3/&2/&1 not authorized to do requested operation.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R3

Change Object Backup List (QEZCHBKL) API

Required Parameter Group:

1	Input structure	Input	Char(*)
2	Input structure length	Input	Binary(4)
3	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Change Object Backup List (QEZCHBKL) API changes the backup type for a list of objects that are specified by the user.

Authorities and Locks

User Index Authority
      *CHANGE

User Index Lock
      *SHRRD

Required Parameter Group

Input structure
    INPUT; CHAR(*)

    This structure includes the keys and data that are needed to make the necessary changes to the backup definitions.

Input structure length
    INPUT; BINARY(4)

    The length of the input structure. A minimum length of 16 is required.

Error code
    I/O; CHAR(*)

    The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Format for Variable Length Records

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Number of variable length records
These fields repeat for each variable length record.		BINARY(4)	Length of variable length record
		BINARY(4)	Key
		BINARY(4)	Length of data
		CHAR(*)	Data

If the length of the data is longer than the key field's data length, the data is truncated at the right. No message is issued.

If the length of the data is smaller than the key field's data length, the data is padded with blanks at the right. No message is issued.

It is not an error to specify a key more than once. If duplicate keys are specified, the last specified value for that key is used.

Each variable length record must be 4-byte aligned. If not, unpredictable results may occur.

Field Descriptions

Number of variable length records. The number of records. Only specific attributes can be changed. Refer to "Valid Keys" on page 13 for more information.

Length of variable length record. The length of each record. Only specific attributes can be changed. Refer to "Valid Keys" on page 13 for more information.

Key. The key specifies either the library or folder attribute. For the list of valid keys, see "Valid Keys" on page 13.

Length of data. The length of the data that is used to specify the value for the given parameter.

Data. The data that is used to specify the value for the given key.

Valid Keys

The following table lists the valid keys for the key field area of the variable length record. For detailed descriptions of the keys, see Field Descriptions.

Key	Type	Field
1	CHAR(*)	Library
2	CHAR(*)	Folder

Field Descriptions

Folder. The backup type selected and the list of folder objects to have their backup type changed. For the format of this field, see "Folder Key Format."

Library. The backup type selected and the list of library objects to have their backup type changed. For the format of this field, see "Library Key Format."

Folder Key Format

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
		CHAR(1)	Backup type
Note: This field repeats for each folder name.			
		CHAR(12)	Folder name

Field Descriptions

Folder name. The folder name of the object to be changed for the backup type that you specified.

Number in array. The number of folder names of objects to have their backup type changed. The value must be 1 or greater.

Backup type. The backup type that you selected for the folder objects. The possible values follow:

1	Back up daily. Back up folder objects during the daily backup. Backing up daily means that the folder objects are also saved on the weekly and monthly backups.
2	Back up weekly. Back up folder objects during the weekly backup. Backing up weekly means that the folder objects are also saved on the monthly backups.
3	Back up monthly. Back up folder objects during the monthly backup.
4	No backup. Folder objects are not backed up at all.

Library Key Format

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
		CHAR(1)	Backup type

Offset		Type	Field
Dec	Hex		
Note: This field repeats for each library name.			
		CHAR(10)	Library name

Field Descriptions

Library name. The library name of the object to be changed for the backup type that you specified.

Number in array. The number of library names of objects to have their backup type changed. The value must be 1 or greater.

Backup type. Backup type that you selected for the library objects. The possible values follow:

1	Back up daily. Back up library objects during the daily backup. Backing up daily means that the library objects are also saved on the weekly and monthly backups.
2	Back up weekly. Backup library objects during the weekly backup. Backing up weekly means that the library objects are also saved on the monthly backups.
3	Back up monthly. Back up library objects during the monthly backup.
4	No backup. Library objects are not backed up at all.

Error Messages

Message ID	Error Message Text
CPF1E65 E	Library backup list in use.
CPF1E6B E	Folder backup list in use.
CPF1EC5 E	Backup option &1 is not valid.
CPF1EEA E	Not authorized to library backup list.
CPF1EEB E	Not authorized to folder backup list.
CPF1E99 E	Unexpected error occurred.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C17 E	Error occurred with input data parameter.
CPF3C81 E	Value for key &1 not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V3R7

Create Media Definition (QSRCRTMD, QsrCreateMediaDefinition) API

Required Parameter Group:

1	Qualified media definition name	Input	Char(20)
2	Input data	Input	Char(*)
3	Length of data	Input	Binary(4)
4	Format name	Input	Char(8)
5	Public authority	Input	Char(10)

6	Text description	Input	Char(50)
7	Replace	Input	Char(1)
8	Error code	I/O	Char(*)

Service Program: QSRLIB01
Default Public Authority: *USE
Threadsafe: No

The Create Media Definition (OPM, QSRCRTMD; ILE, QsrCreateMediaDefinition) API creates a media definition specified by the user. ≫ A **media definition** defines the devices, media, and data format to be used in parallel by a save or restore operation. For more information about using a media definition, see Save to multiple devices in the Backup and recovery topic. ≪

Authorities and Locks

Media Definition Authority
> *OBJMGMT, *OBJEXIST, and *READ. These authorities are required only if an existing media definition is to be replaced.

Library Authority
> *EXECUTE, *ADD and *READ

Media Definition Lock
> *EXCL

Library Lock
> *SHRUPD

Required Parameter Group

Qualified media definition name
> INPUT; CHAR(20)

> The media definition to be created. The first 10 characters contain the media definition name. The second 10 characters contain the name of the library in which the media definition is located.

> You can use the following special value for the library name. It should be noted, however, that the library name that is actually used is not passed back to the user. Care should be taken when using this special value to avoid unexpected results.

CURLIB   The job's current library is used to locate the media definition. If no library is specified as the current library for the job, the QGPL library is used.

Input data
> INPUT; CHAR(*)

> The variable that is to hold all the information defining the use of multiple tape files for a save or restore operation. See "Input Data Format" on page 16 for the format of the input data.

Length of data
> INPUT; BINARY(4)

> The length of the data in the input data parameter. The length of data parameter may be specified up to the size of the input data variable specified in the user program. If the length of data parameter specified is larger than the allocated size of the input data variable specified in the user program, the results are not predictable. ≫ The minimum length is 72 bytes for format TAPE0100 and 92 bytes for format TAPE0200. ≪

Format name
> INPUT; CHAR(8)

The name of the format for input data. The valid values are:

TAPE0100   Tape devices and media
≫ *TAPE0200*   Tape devices and media (extended) ≪

Public authority
>INPUT; CHAR(10)

>The authority you give to users who do not have specific private or group authority to the media definition. Once the media definition has been created, its public authority stays the same when it is moved to another library or restored from backup media.

>If the replace parameter is used and an existing media definition is replaced, this parameter is ignored. All authorities are transferred from the replaced media definition to the new one. The valid values for this parameter are:

*ALL	The user can perform all authorized operations on the media definition.
Authorization list name	The media definition is secured by the specified authorization list, and its public authority is set to *AUTL. The specified authorization list must exist on the system when this API is issued. If the list does not exist, the create process fails, and an error message is returned to the application.
*CHANGE	The user has read, add, update, and delete authority for the media definition and can read the object description.
*EXCLUDE	The user cannot access the media definition in any way.
*LIBCRTAUT	The public authority for the media definition is taken from the CRTAUT value for the target library when the object is created. If the CRTAUT value for the library changes later, that change does not affect media definitions already created. If the CRTAUT value contains an authorization list name and that authorization list secures an object, do not delete the list. If you do, the next time you call this API with the *LIBCRTAUT parameter, it will fail.
*USE	The user can read the object description and contents, but cannot change the media definition.

Text description
>INPUT; CHAR(50)

>A brief description of the media definition.

Replace
>INPUT; CHAR(1)

>Whether you want to replace an existing media definition. The valid values are:

0	Do not replace an existing media definition of the same name and library.
1	Replace an existing media definition of the same name and library. The replaced media definition is moved to the QRPLOBJ library, which is cleared at system IPL. For details about authorities, ownership, and renaming, see the discussion of the REPLACE parameter in Control Language (CL)

Error code
>I/O; CHAR(*)

>The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Input Data Format

The input data consists of a header and a set of device definitions and media file definitions. The following defines the format for the header. For detailed descriptions of the fields, see "Field Descriptions for Input Data" on page 17.

Format TAPE0100

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Reserved
4	4	BINARY(4)	Reserved
8	8	BINARY(4)	Maximum parallel device resources
12	C	BINARY(4)	Minimum parallel device resources
16	10	BINARY(4)	Offset to first device definition
20	14	BINARY(4)	Number of device definitions
		CHAR(*)	Device definitions

Format TAPE0200

Offset			
Dec	**Hex**	**Type**	**Field**
0	0		All fixed fields from format TAPE0100
24	18	BINARY(4)	Length of header
28	1C	BINARY(4)	Device allocation
32	20	BINARY(4)	Save format
		CHAR(*)	Device definitions

《

Field Descriptions for Input Data

》 **Device allocation.** When to allocate the tape devices. This field is ignored for save operations that specify a target release earlier than V5R4M0. The default value is 0. The possible values are:

0   All tape devices are allocated at the beginning of the operation.
1   One tape device is allocated at the beginning of a save operation. Additional devices are allocated when data is ready to be written, at which time the number of devices specified for the Minimum parallel device resources field is required.
2   The number of devices specified for the Minimum parallel device resources field is allocated at the beginning of a save operation. Additional devices are allocated when data is ready to be written.

《

Device definitions. A description of the devices to be used. See "Device Definition Format" on page 18 for the format of a device definition.

》 **Length of header.** The length of the fixed portion of the header information. The value must be 36. 《

Maximum parallel device resources. The maximum number of device resources to use in parallel. The possible values are 0 through 32. If 0 is specified, the value assumed is the total number of media file definitions specified in all of the device definitions.

Minimum parallel device resources. The minimum number of device resources to use in parallel. A save or restore operation will end if fewer resources are available. A restore operation will also end if any of the devices specified have no resources available. The possible values are 0 through 32. If 0 is specified, the value assumed is the number of device definitions specified.

Number of device definitions. The number of device definitions for the media definition. The possible values are 1 through 32.

Offset to first device definition. The offset from the beginning of the input data to the first device definition for the media definition. This value must be a multiple of 4.

Reserved. The value must be hexadecimal zeros.

»Save format. Whether to save data in serial format or parallel format. This field is ignored for restore operations. The default value is -2. The possible values are:

-2   If one library is saved, it is saved in parallel format. If more than one library is saved, all libraries are saved in serial format.
-1   All data is saved in serial format.
0   All data is saved in parallel format.

«

Device Definition Format

Format TAPE0100

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Offset to next device definition
4	4	CHAR(10)	Device name
14	E	CHAR(2)	Reserved
16	10	BINARY(4)	Offset to first media file definition
20	14	BINARY(4)	Number of media file definitions
		CHAR(*)	Media file definitions

Format TAPE0200

Offset			
Dec	**Hex**	**Type**	**Field**
0	0		All fixed fields from format TAPE0100
24	18	BINARY(4)	Length of device definition
		CHAR(*)	Media file definitions

«

Field Descriptions for Device Definition

Device name. The name of a tape device description or tape media library device description.

»Length of device definition. The length of the fixed portion of the device definition. The value must be 28. «

Media file definitions. A description of the media files to be used on this device. See "Media File Definition Format" on page 19 for the format of a media file definition.

Number of media file definitions. The number of media file definitions for the device. The possible values are 1 through 32.

Offset to first media file definition. The offset from the beginning of the input data to the first media file definition for the device. This value must be a multiple of 4.

Offset to next device definition. The offset from the beginning of the input data to the next device definition for the media definition. This value must be a multiple of 4.

Reserved. The value must be hexadecimal zeros.

Media File Definition Format

Format TAPE0100

| Offset | | | |
Dec	Hex	Type	Field
0	0	BINARY(4)	Offset to next media file definition
4	4	BINARY(4)	Sequence number
8	8	BINARY(4)	Offset to volume identifier array
12	C	BINARY(4)	Number of volume identifiers
16	10	BINARY(4)	Length of volume identifier
20	14	BINARY(4)	Starting volume array element
		CHAR(*)	Volume identifier array

Format TAPE0200

| Offset | | | |
Dec	Hex	Type	Field
0	0		All fixed fields from format TAPE0100
24	18	BINARY(4)	Length of media file definition
		CHAR(*)	Volume identifier array

$\gg$

Field Descriptions for Media File Definition

$\gg$ **Length of media file definition.** The length of the fixed portion of the media file definition. The value must be 28. $\ll$

Length of volume identifier. The number of bytes in each volume identifier. The possible values are 0 through 6. If 0 is specified, the number of volume identifiers specified must be 0.

Number of volume identifiers. The number of volume identifiers used for the tape file. The possible values are 0 through 75. If 0 is specified, the volume currently placed in the device is used. If 0 is specified for a tape media library device, volume identifiers must be supplied by using the Tape Management exit program during the save or restore operation.

Offset to next media file definition. The offset from the beginning of the input data to the next media file definition for the device. This value must be a multiple of 4.

Offset to volume identifier array. The offset from the beginning of the input data to the first volume identifier for the tape file. This value must be a multiple of 4.

Sequence number. The tape file sequence number for the media file.

The possible values are:

0	A save operation begins after the last sequence number on the starting volume. A restore operation searches the starting volume for a media file containing any of the objects to restore.
1-16777215	The sequence number of the tape file.

Starting volume array element. The element in the volume identifier array containing the volume on which the save or restore operation should begin. The possible values are 0 through the number of volume identifiers specified. If the number of volume identifiers is 0, this value must be 0. If the number of volume identifiers is greater than 0, this value must be greater than 0.

Volume identifier array. An array of volume identifiers. The save or restore operation will use the volumes in the order specified, beginning with the starting volume array element. If additional volumes are needed after the last array element is used, the save or restore operation will call the Tape Management exit program or prompt the user to provide each additional volume. The possible value for a volume identifier is:

Volume identifier   The identifier of a volume.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF386F E	Value in input data parameter not valid.
CPF3C17 E	Error occurred with input data parameter.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF9800 E	All CPF98xx messages could be signaled. xx is from 01 to FF.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V4R4

Delete Media Definition (QSRDLTMD, QsrDeleteMediaDefinition) API

Required Parameter Group:

1	Qualified media definition name	Input	Char(20)
2	Error code	I/O	Char(*)

Service Program: QSRLIB01
Default Public Authority: *USE
Threadsafe: No

The Delete Media Definition (OPM, QSRDLTMD; ILE, QsrDeleteMediaDefinition) API deletes a media definition specified by the user. ≫ A **media definition** defines the devices, media, and data format to be used in parallel by a save or restore operation. For more information about using a media definition, see Save to multiple devices in the Backup and recovery topic. ≪

Authorities and Locks

Media Definition Authority
     *OBJEXIST

Library Authority
     *EXECUTE

Media Definition Lock
     *EXCL

Library Lock
     *SHRUPD

Required Parameter Group

Qualified media definition name
     INPUT; CHAR(20)

     The media definition to be deleted. The first 10 characters contain the media definition name. The second 10 characters contain the name of the library in which the media definition is located.

     The media definition name can be either a specific name or a generic name, which is a string of one or more characters followed by an asterisk (*). If you specify a generic name, this API deletes all media definitions that have names beginning with the string for which the user has authority.

     You can use the following special values for the library name. It should be noted, however, that the library name that is actually used is not passed back to the user. Care should be taken when using these special values to avoid unexpected results.

*CURLIB	The job's current library is used to locate the media definition. If no library is specified as the current library for the job, the QGPL library is used.
*LIBL	The library list is used to locate the media definition.
*USRLIBL	The user portion of the job's library list is used to locate the media definition.
*ALL	All libraries in the system, including QSYS, are searched.
*ALLUSR	All user-defined libraries, plus libraries containing user data and having names starting with Q. For information on the libraries included, see *ALLUSR in Generic library names.

Error code
     I/O; CHAR(*)

     The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF9800 E	All CPF98xx messages could be signaled. xx is from 01 to FF.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V4R4

Dump Device (QTADMPDV) API

Required Parameter Group:

1	Device name	Input	Char(10)

Optional Parameter Group:

2	Type of information	Input	Char(10)
3	Problem identifier	Input	Char(10)
4	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Dump Device (QTADMPDV) API collects information for your IBM service representative. This API should be used immediately after a suspected device and/or tape management system failure. If the API is not used immediately, other device operations may cause the flight recorders to wrap, which could result in lost information. A problem identifier will be created and an APAR library will be generated similar to the Save APAR Data (SAVAPARDTA) command. To save the APAR library, use Work with Problem (WRKPRB) command. Choose the option to work with the problem and then the option to save the APAR library. If an existing problem identifier is passed to this API, then the spooled files generated will be logged against that problem identifier and no new problem identifier will be generated.

The Dump Device API currently supports the following device types:
- Tape (TAP) devices
- Tape media library (TAPMLB) devices
- Optical (OPT) devices
- Optical media library (OPTMLB) devices

Note: The information provided in and the number of spooled files may change at anytime. The information provided is intended for problem determination.

The Dump Device (QTADMPDV) API dumps the following information for tape or tape media library devices, into spooled files:
- ≫ Dump of the device description for the device specified in the parameter that is passed to the program.
- Device description details of the device.
- Device capabilities (QTARDCAP output).
- QTARDINF API output.
- Licensed Internal Code tape flight recorders.
- IOP trace for the device.
- QSYSOPR message queue.
- Message queue for the user/device doing the QTADMPDV call.
- Licensed Internal Code logs from the last 24 hours.
- Product Activity Logs from the last 24 hours.
- The PTF list.
- A Work with Configuration Status (WRKCFGSTS) listing.
- Media and Storage Extensions (MSE) flight recorder.

- The history log (QHST).
- QTAPARB job log and display job (DSPJOB) information.
- Job logs and display jobs (DSPJOB) of the active jobs that have used the device as indicated in the flight recorder data.
- The job log and display job (DSPJOB) of the job that is processing this API.
- The Media library inventory information and display file description (DSPFD).
- The Media library category information and display file description (DSPFD).
- The Media library filter information and display file description (DSPFD).
- The Display Hardware Resources output (DSPHDWRSC).
- QTAHRSRV flight recorders.
- The virtual tape information area.
- The problem log summary.
- Communication information that is associated with the media library device. This includes the line, controller, and device descriptions. 《

Note that this API will generate multiple spooled files that may get large depending upon the job logs that are being printed and the size of the other device information. Submitting the call to batch may be used if system performance is a concern. That is, if the API is called from the system console at high priority, it may degrade performance on other critical processing. Since many and potentially large spooled files may be generated, ensure that there is enough system storage available to handle the request.

Authorities and Locks

》

Device description: *USE

To dump the IOP trace you need use (*USE) authority to the Control Device API (QTACTLDV). *See the documentation for this API for additional authority requirements.*

To dump the PTF details you need use (*USE) authority to the Display Program Temporary Fix (DSPPTF) command. *See the documentation for this command for additional authority requirements.*

To dump the VLOGS you need use (*USE) authority to the Print Internal Data (PRTINTDTA) command and Service (*SERVICE) special authority. *See the documentation for this command for additional authority requirements.*

To dump the Product Activity Logs (PALS) you need use (*USE) authority to the Print Error Log (PRTERRLOG) command. *See the documentation for this command for additional authority requirements.*

To dump the QTAPARB job log and display job information, you need Job Control (*JOBCTL) and All Object (*ALLOBJ) special authorities.

To dump the Media Library Inventory file description, you need use (*USE) authority to the QATAMID and QLTAMID files in QUSRSYS.

To dump the Media Library Category file description, you need use (*USE) authority to the QATACGY and QLTACGY files in QUSRSYS.

To dump the Cartridge Filter file description, you need use (*USE) authority to the QATAFTR file in QUSRSYS.

To dump the QTAHRSRV flight recorders you need use (*USE) authority to the Dump System Object (DMPSYSOBJ) command. *See the documentation for this command for additional authority requirements.*

≫

Required Parameter

Device name
INPUT; CHAR(10)

The name of the device for which debugging information is being dumped.

Optional Parameter Group

Type of information
INPUT; CHAR(10)

The type of information to be dumped.
Valid values are:

*ALL	All information needed by IBM will be dumped to spooled files.
*MSE	Media and Storage Extension (MSE) flight recorder will be dumped.
	≫ This is only valid for tape or tape media library devices. ≪

Problem identifier
INPUT; CHAR(10)

The problem identifier of the problem being analyzed. Problems with different system origins can have the same identifier. The possible values are:

*NEW	A problem identifier will be created.
problem-identifier	The 10-character problem identifier of the problem being selected.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter. If this parameter is omitted, diagnostic and escape messages are issued to the application.

Examples

The following are examples of calls to the API from command entry:

- CALL QTADMPDV TAP01

  The dump device will dump information about TAP01 and assigns it to a created problem identifier.

- CALL QTADMPDV TAPMLB01

  The dump device will dump information about TAPMLB01 and assigns it to a created problem identifier.

- CALL QTADMPDV (TAP01 *ALL 9628851615 x'00000000')

  The dump device will dump information about TAP01 and assign it to an existing problem identifier.

Error Messages

Message ID	Error Message Text
CPF3C90 E	Literal value cannot be changed.
CPF6709 E	Parameter &3 not correct.
CPF6721 E	Device &1 not a tape device.
CPF673F E	Device &1 does not support &2.
CPF9814 E	Device &1 not found.
CPF9825 E	Not authorized to device &1.

Message ID	Error Message Text
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R1

Free Object (QTAFROBJ) API

Required Parameter Group:

1	Request variable	Input	Char(*)
2	Length of request variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Free Object (QTAFROBJ) API will "suspend" a document object specified by the caller of the API. A call to this API forces the system storage that is occupied by the data portion of the specified object to be freed. Only the data portion of the objects is freed, not the descriptions of the object. This function is similar to the save with storage freed option, STG(*FREE), on the Save Document Library Object (SAVDLO) command.

The caller of this API is required to verify that the specified object has not been changed since it was last saved.

Notes:

1. To use this API, you need the Media and Storage Extensions feature of the i5/OS.

2. For a document of type *DOC, the caller must be enrolled in the system distribution directory to use this API.

Authorities and Locks

Object Authority
    *CHANGE

    *OBJEXIST

Directory Authority
    *X

Required Parameter Group

Request variable
    INPUT; CHAR(*)

    The request variable that identifies the object to be suspended.

Length of request variable
    INPUT; BINARY(4)

    The length of the request variable provided. Valid values range from 48 through 32048.

Format name
    INPUT; CHAR(8)

The format of the object information being passed to the QTAFROBJ API. The TAFO0100 format must be used for this API. See "TAFO0100 Format" to view the object information required to perform this API.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

TAFO0100 Format

The following table shows the object information that is required for the TAFO0100 format. For more details about the fields in the following table, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	CHAR(10)	Object name
10	A	CHAR(10)	Object library
20	14	CHAR(10)	Reserved
30	1E	CHAR(10)	Object type
40	28	BINARY(4)	Offset to path name structure
44	2C	BINARY(4)	Length of path name structure
		CHAR(*)	Path name structure

Field Descriptions

Length of the path name structure. The length, in bytes, of the path name structure. This field must be set to zero if the object does not have a path name structure passed. Valid values are 0 and 48 through 32048.

Object library. The library name of the object to be freed. The special value is:

*PATH   The path name structure contains the object information.

Object name. The name of the object to be freed by the API. The special value is:

*PATH   The path name structure contains the object information.

Object type. The type of object specified to be freed by the API. Possible values follow:

*DOC   The object to be suspended is a document.
*PATH   The path name structure will contain the object information.

Offset to path name structure. The offset from the start of the structure, in bytes, to a path name structure that contains object path name and translation information. This field must be set to zero if the object does not have a path name structure. Valid values are 0 and 48 through 32048.

Path name structure. The path name structure and translation information for the suspended object. The path name structure contains information such as CCSID, country or region, and language. For more information on this structure, see Path name format. The path name must be in the library file system format; for example, /QSYS.LIB/QDOC.LIB/DOC1.DOC

Reserved. An ignored field. This field must be set to blanks.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3C4B E	Value not valid for field &1.
CPF3C4C E	Value not valid for field &1.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF6708 E	Command ended due to error.
CPF67C2 E	Path name structure field &7 not valid.
CPF67C4 E	Object &1 type &2 in library &3 not freed. Object in use.
CPF67C5 E	Object &1 type &2 in library &3 not freed.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R3

List Save File (QSRLSAVF) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Format name	Input	Char(8)
3	Qualified save file name	Input	Char(20)
4	Object name filter	Input	Char(10)
5	Object type filter	Input	Char(10)
6	Continuation handle	Input	Char(36)
7	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The List Save File (QSRLSAVF) API lists the contents of a save file. The generated list replaces any data that already exists in the user space; it does not add the new list to an existing one. The generated list is not sorted.

Authorities and Locks

Save File Library Authority
    *USE

Save File Authority
    *USE

Save File Lock
    *EXCLRD

User Space Authority
    *CHANGE

User Space Library Authority
> *EXECUTE

User Space Lock
> *EXCLRD

Required Parameter Group

Qualified user space name
> INPUT; CHAR(20)

> The user space that is to receive the created list. The first 10 characters contain the user space name, and the second 10 characters contain the name of the library where the user space is located. You can use these special values for the library name:

*CURLIB	The job's current library
*LIBL	The library list

Format name
> INPUT; CHAR(8)

> The content and format of the information returned for the save file. The possible format names are:

SAVF0100	Library level
SAVF0200	Object level
SAVF0300	Member level
≫ SAVF0400	Spooled files ≪

> For more information, see the specified formats in the "Format of the Generated List" on page 29.

Qualified save file name
> INPUT; CHAR(20)

> The save file about which to list information, and the library in which the save file is located. The first 10 characters contain the save file name, and the second 10 characters contain the library name. You can use these special values for the library name:

*CURLIB	The job's current library
*LIBL	The library list

Object name filter
> INPUT; CHAR(10)

> The name of the objects to search for. This name may be a simple name, a generic name, or the special value *ALL. If the name is not a valid name, an empty list will be returned. This field must be *ALL for the SAVF0100 format.

Object type filter
> INPUT; CHAR(10)

> The type of objects to search for. You may either enter a specific type or the special value *ALL. For a complete list of the available object types, see Control Language (CL) information. This field must be *ALL for the SAVF0100 format, the SAVF0300 format, ≫ and the SAVF0400 format. ≪

Continuation handle
> INPUT; CHAR(36)

> The handle used to continue from a previous call to this API that resulted in partially complete information. You can determine if a previous call resulted in partially complete information by

checking the information status field in the generic user space header following the API call. For information about the generic header, see User space format for list APIs.

If the API is not attempting to continue from a previous call, this parameter must be set to blanks. Otherwise, a valid continuation value must be supplied. The value may be obtained from the continuation handle returned field in the header section. See "Format of the Generated List" for information about the header section.

Error code
> I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Format of the Generated List

The save file list consists of:

* A user area
* A generic header
* An input parameter section
* A header section
* A list data section (containing one of the following):
    * SAVF0100 format
    * SAVF0200 format
    * SAVF0300 format
    * ≫ SAVF0400 format≪

For details about the user area and generic header, see User space format for list APIs. For details about the remaining items, see the following sections. For detailed descriptions of the fields in the list returned, see "Field Descriptions" on page 31.

When you retrieve list entry information from a user space, you must use the entry size returned in the generic header. The size of each entry may be padded at the end. If you do not use the entry size, the result may not be valid. For examples of how to process lists, see the DLTOLDSPLF example programs in API Examples.

Input Parameter Section

Offset			
Dec	Hex	Type	Field
0	0	CHAR(10)	User space name specified
10	A	CHAR(10)	User space library name specified
20	14	CHAR(8)	Format name
28	1C	CHAR(10)	Save file name specified
38	26	CHAR(10)	Save file library name specified
48	30	CHAR(10)	Object name filter specified
58	3A	CHAR(10)	Object type filter specified
68	44	CHAR(36)	Continuation handle specified

Header Section

Offset			
Dec	Hex	Type	Field
0	0	CHAR(10)	User space name used
10	A	CHAR(10)	User space library name used
20	14	CHAR(10)	Save file name used
30	1E	CHAR(10)	Save file library name used
40	28	CHAR(36)	Continuation handle returned

SAVF0100 Format

Offset			
Dec	Hex	Type	Field
0	0	CHAR(10)	Library saved
10	A	CHAR(10)	Save command
20	14	CHAR(8)	Save date and time
28	1C	BINARY(4)	Auxiliary storage pool
32	20	BINARY(4)	Records
36	24	BINARY(4)	Objects saved
40	28	BINARY(4)	Access paths
44	2C	CHAR(10)	Save active
54	36	CHAR(6)	Release level
60	3C	CHAR(1)	Data compressed
61	3D	CHAR(8)	System serial number
69	45	CHAR(3)	Reserved
72	48	CHAR(10)	Auxiliary storage pool device name
≫ 82	52	CHAR(2)	Reserved
84	54	BINARY(4)	Members in library saved
88	58	BINARY(4)	Spooled files saved ≪

SAVF0200 Format

Offset			
Dec	Hex	Type	Field
0	0	CHAR(10)	Object name
10	A	CHAR(10)	Library saved
20	14	CHAR(10)	Object type
30	1E	CHAR(10)	Extended object attribute
40	28	CHAR(8)	Save date and time
48	30	BINARY(4)	Object size

Offset			
Dec	Hex	Type	Field
52	34	BINARY(4)	Object size multiplier
56	38	BINARY(4)	Auxiliary storage pool
60	3C	CHAR(1)	Data saved
61	3D	CHAR(10)	Object owner
71	47	CHAR(20)	Document library object (DLO) name
91	5B	CHAR(63)	Folder
154	9A	CHAR(50)	Text description
204	CC	CHAR(10)	Auxiliary storage pool device name

SAVF0300 Format

Offset			
Dec	Hex	Type	Field
0	0	CHAR(10)	File name
10	A	CHAR(10)	Library saved
20	14	CHAR(10)	Member name
30	1E	CHAR(10)	Extended object attribute
40	28	CHAR(8)	Save date and time
48	30	BINARY(4)	Members saved

SAVF0400 Format

Offset			
Dec	Hex	Type	Field
0	0	CHAR(10)	Job name
10	A	CHAR(10)	User name
20	14	CHAR(6)	Job number
26	1A	CHAR(10)	Spooled file name
36	24	BINARY(4)	Spooled file number
40	28	CHAR(8)	Job system name
48	30	CHAR(7)	Creation date
55	37	CHAR(6)	Creation time
61	3D	CHAR(10)	Output queue name
71	47	CHAR(10)	Output queue library
81	51	CHAR(3)	Reserved ≪

Field Descriptions

Access paths. The number of logical file access paths that were saved for the library.

Auxiliary storage pool. The auxiliary storage pool (ASP) of the object when it was saved. For the SAVF0100 format, this is the ASP of the library. For the SAVF0200 format, this is the ASP of the object. The possible values are:

1	System ASP
2 - 32	Basic user ASPs
33 - *255*	Independent ASPs

Auxiliary storage pool device name. The name of the independent auxiliary storage pool (ASP) device of the object when it was saved. For the SAVF0100 format, this is the ASP of the library. For the SAVF0200 format, this is the ASP of the object.

≫ **Creation date.** The date the spooled file was created, in the format CYYMMDD:

C	Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
YY	Year
MM	Month
DD	Day

Creation time. The time the spooled file was created, in the format HHMMSS:

HH	Hour
MM	Minute
SS	Second ≪

Continuation handle returned. A continuation point for the API.

This value is set based on the contents of the information status variable in the generic header for the user space. The following situations can occur:

- Information status-C. The information returned in the user space is valid and complete. No continuation is necessary and the continuation handle is set to blanks.
- Information status-P. The information returned in the user space is valid but incomplete. The user may call the API again, continuing where the last call ended. The continuation handle contains a value that may be supplied as an input parameter in later calls.
- Information status-I. The information returned in the user space is not valid or complete. The contents of the continuation handle are unpredictable.

Continuation handle specified. The handle used to continue from a previous call to this API that resulted in partially complete information.

Data compressed. Whether the data was stored in compressed format. The possible values are:

0	The data is not compressed.
1	The data is compressed.

Data saved. Whether the data for this object was saved with the object. The possible values are:

0	The data was not saved. The object's storage was freed by a previous save command before this save operation.
1	The data was saved. The object's storage was not freed by a previous save command before this save operation.

Document library object (DLO) name. The name of the document, folder, or mail object that was saved. If the object is a document or folder, the first 12 characters will contain the DLO name. If the object is a mail object, the full 20 characters will be used for the mail object name. If the save file does not contain DLO information, this field will be blank.

Extended object attribute. Extended information about the object type. If there is not an extended object attribute for the object, this field will be blank.

File name. The name of the file that was saved.

Folder. The name of the folder that was saved. The folder name is a fully qualified name. If the object is not a *FLR or *DOC object, this field will be blank. For *DOC and *FLR objects, this field will be set to the qualified name of the folder or to *NONE.

Format name. The format of the returned output.

» Job name. The name of the job that owns the spooled file.

Job number. The number of the job that owns the spooled file.

Job system name. The name of the system where the job that owns the spooled file ran. «

Library saved. The name of the library from which the objects are saved.

» Members in library saved. The number of members saved for the library. «

Member name. The name of the file member that is saved. The member names are not in sorted order.

Members saved. The number of members saved for the file.

Object name. The name of the object saved. If the object is a DLO object, this field will contain the system name of the object.

Object name filter specified. The name of the objects to search for. Only objects with names that match the filter are listed.

Object owner. The name of the object owner's user profile.

Objects saved. The number of objects that are saved for this library.

Object size. The size of the object in units of the size multiplier. The true object size is equal to or smaller than the object size multiplied by the object size multiplier.

Object size multiplier. The value to multiply the object size by to get the true size. The value is 1 if the object is smaller than or equal to 999 999 999 bytes, 1024 if it is larger than 999 999 999 but smaller than or equal to 4 294 967 295, and 4096 if larger than 4 294 967 295.

Object type. The type of object. For a list of object types, see Control Language (CL) information.

Object type filter specified. The type of objects to search for. Only object types that match the filter are listed.

» Output queue library. The name of the output queue library that contained the spooled file.

Output queue name. The name of the output queue that contained the spooled file. «

Records. The number of records used to contain the saved information in the save file.

Release level. The earliest release level of the operating system on which the objects can be restored.

Reserved. An ignored field.

Save active. Whether objects in the library are allowed to be updated while they are being saved. The possible values are:

*LIB	Objects in the library are saved while in use by another job. All of the objects in the library reached a checkpoint together and were saved in a consistent state in relationship to each other. All objects in the library are saved at the same time.
*NO	Objects in the library are not saved while in use by another job.
*SYNCLIB	Objects in the library are saved while in use by another job. All of the objects and all of the libraries in the save operation reached a checkpoint together. The objects and the libraries were saved in a consistent state in relationship to each other.
*SYSDFN	Objects in the library are saved while in use by another job. Objects in the library may have reached a checkpoint at different times and may not be in a consistent state in relationship to each other.
*YES	Document library objects are saved while in use by another job. This value is valid only if the SAVDLO command is used for the save operation.

Save command. The save command that is used when the save operation is performed. The possible values are:

QSYS	Contents of the save file are created by the operating system by using a function other than the CL commands.
SAVCFG	Saves configuration information.
SAVCHGOBJ	Saves objects that changed since the date and time specified on the referenced date parameter.
SAVDLO	Saves documents or folders located in library QDOC.
SAVLIB	Saves a copy of a library.
SAVLICPGM	Saves licensed programs.
SAVOBJ	Saves an object or group of objects from the same library.
SAVSECDTA	Saves objects required for the security function.

Save date and time. The time at which the objects were saved in system time-stamp format.

Save file library name specified. The name of the save file library as specified in the call to the API.

Save file library name used. The name of the save file library used to produce the listing.

Save file name specified. The name of the save file as specified in the call to the API.

Save file name used. The name of the save file used to produce the listing.

≫ **Spooled file name.** The name of the spooled file.

Spooled file number. The number of the spooled file in the job that owns it.

Spooled files saved. The number of spooled files saved in the save file. ≪

System serial number. The serial number of the system on which the save was performed. If the save media is from a System/38$^{(TM)}$, the system serial number will be blank.

Text description. The text description of the object. If the object is a DLO object, the following pertains:

- Characters 1 through 44 contain the text description.
- The last 6 characters are padded with blanks.

≫ **User name.** The name of the user who owns the spooled file. ≪

User space library name specified. The name of the library containing the user space as specified in the call to the API.

User space library name used. The name of the library used to produce the listing.

User space name specified. The name of the user space as specified in the call to the API.

User space name used. The name of the user space used to produce the listing.

Error Messages

Message ID	Error Message Text
CPF22FD E	Continuation handle not valid for API &1.
CPF24B4 E	Severe error while addressing parameter list.
CPF3704 E	Request ended; data management error occurred.
CPF3743 E	File cannot be restored, displayed, or listed.
CPF3782 E	File &1 in &2 not a save file.
CPF3793 E	Machine storage limit reached.
CPF381F E	Save file &1 cannot be processed by API QSRLSAVF.
CPF3812 E	Save file &1 in &2 in use.
CPF3C21 E	Format name &1 is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3C90 E	Literal value cannot be changed.
CPF8100 E	All CPF81xx messages could be returned. xx is from 01 to FF.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9806 E	Cannot perform function for object &2 in library &3.
CPF9807 E	One or more libraries in library list deleted.
CPF9808 E	Cannot allocate one or more libraries on library list.
CPF9809 E	Library &1 cannot be accessed.
CPF9810 E	Library &1 not found.
CPF9812 E	File &1 in library &2 not found.
CPF9820 E	Not authorized to use library &1.
CPF9822 E	Not authorized to file &1 in library &2.
CPF9830 E	Cannot assign library &1.
CPF9838 E	User profile storage limit exceeded.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V2R3

Open List of Objects to be Backed Up (QEZOLBKL) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	List information	Output	Char(80)

4	Number of records to return	Input	Binary(4)
5	Format name	Input	Char(8)
6	Object type	Input	Char(10)
7	Type of backup to select	Input	Char(10)
8	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Open List of Objects to be Backed Up (QEZOLBKL) API retrieves an open list of the objects that are to be backed up.

For more information, see Process Open List APIs.

Authorities and Locks

User Index Authority
      *USE

User Index Lock
      *SHRRD

Required Parameter Group

Receiver variable
      OUTPUT; CHAR(*)

      The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
      INPUT; BINARY(4)

      The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable.

List information
      OUTPUT; CHAR(80)

      Information about the list that is created by this program. For a description of the layout of this parameter, see "Format of List Information" on page 39.

Number of records to return
      INPUT; BINARY(4)

      The number of records in the list to put into the receiver variable. The value must be 0 or greater.

Format name
      INPUT; CHAR(8)

      The name of the format to be used to return the requested information. One of the following format names may be used:

OBKL0100	Library basic information format. The object type parameter must be *LIB. For more information, see "OBKL0100 Format" on page 37.
OBKL0200	Folder basic information format. The object type parameter must be *FLR. For more information, see "OBKL0200 Format" on page 37.
OBKL0600	Complete information format. The object type parameter may be *FLR or *LIB. For more information, see "OBKL0600 Format" on page 38.

Object type
> INPUT; CHAR(10)

> The type of the objects to be returned in the list. One of the following object types may be used:

FLR   The folder information is returned. The format name may be OBKL0200 or OBKL0600.
LIB   The library information is returned. The format name may be OBKL0100 or OBKL0600.

Type of backup to select
> INPUT; CHAR(10)

> The backup type of the objects that you request. Allowable backup types follow:

DAILY   Return information for objects that are backed up *DAILY.
WEEKLY   Return information for objects that are backed up *WEEKLY.
MONTHLY   Return information for objects that are backed up *MONTHLY.
ALL   Return information for objects that are backed up *DAILY, *WEEKLY, or *MONTHLY.

Error code
> I/O; CHAR(*)

> The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Format of Receiver Variable

The following tables describe the order and format of the data that is returned in the receiver variable.

OBKL0100 Format

The OBKL0100 format includes the basic information for a library object entry. The following table shows how this information is organized. For detailed descriptions of the fields in the list, see "Field Descriptions" on page 38.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(10)	Backup option
10	A	CHAR(10)	Library name
20	14	CHAR(2)	Reserved

OBKL0200 Format

The OBKL0200 format includes the basic information for a folder object entry. The following table shows how this information is organized. For detailed descriptions of the fields in the list, see "Field Descriptions" on page 38.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(10)	Backup option
10	A	CHAR(12)	Folder name

OBKL0600 Format

The OBKL0600 format includes the complete information for a library or folder object entry. The following table shows how this information is organized. For detailed descriptions of the fields in the list, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0		Returns everything from format OBKL0100 or OBKL0200
22	16	CHAR(7)	Last backup date
29	1D	CHAR(6)	Last backup time
35	23	CHAR(50)	Object description text
85	55	CHAR(1)	Changed since last backup
86	56	CHAR(21)	Reserved

Field Descriptions

Backup option. The backup option defined for the object. The possible values follow:

*DAILY	Daily backup option.
*WEEKLY	Weekly backup option.
*MONTHLY	Monthly backup option.

Changed since last backup. Whether the object has changed since the last backup. The possible values follow:

0	No change has been made since the last backup.
1	A change has been made since the last backup.

Folder name. The folder name of the object to be backed up.

Last backup date. The date that the object was last backed up. The format of this field is in the CYYMMDD as follows:

C	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
YY	Year
MM	Month
DD	Day

Last backup time. The time that the object was last backed up. The format of this field is in the HHMMSS as follows:

HH	Hour
MM	Minute
SS	Second

Library name. The library name of the object to be backed up.

Object description text. The text that describes the object.

Reserved. An ignored field.

Format of List Information

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Total records
4	4	BINARY(4)	Records returned
8	8	CHAR(4)	Request handle
12	C	BINARY(4)	Record length
16	10	CHAR(1)	Information complete indicator
17	11	CHAR(13)	Date and time created
30	1E	CHAR(1)	List status indicator
31	1F	CHAR(1)	Reserved
32	20	BINARY(4)	Length of information returned
36	24	BINARY(4)	First record in buffer
40	28	BINARY(4)	Authority reason code
44	2C	CHAR(36)	Reserved

Field Descriptions

Authority reason code. Whether all information that you requested has been supplied due to the user's authority. The list of folders or libraries may not be the complete list of objects on the system if the user does not have the required authority to list the object. A user with *SAVSYS or *ALLOBJ authority is able to get a complete list. Without this authority, the list may not be complete.

0000	The list is complete.
0001	The list may be partial due to authorization.

Date and time created. The date and time that the list was created. The 13 characters follow:

1	Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
2-7	The date, in YYMMDD (year, month, and day) format.
8-13	The time of day, in HHMMSS (hours, minutes, and seconds) format.

First record in buffer. The number of the first record in the receiver variable.

Information complete indicator. Whether all information that was requested has been supplied.

I	Incomplete information. An interruption causes the list to contain incomplete information about a buffer or buffers.
P	Partial and accurate information. Partial information is returned when the maximum space is used and not all of the buffers that were requested are read.
C	Complete and accurate information. All the buffers that were requested are read and returned.

Length of information returned. The size, in bytes, of the information that is returned in the receiver variable.

List status indicator. The status of building the list.

0	Building the list is pending.

1	The list is in the process of being built.
2	The list has been completely built.
3	An error occurred when the system built the list. The next call to the Get List Entries (QGYGTLE) API causes the error to be signaled to the caller of QGYGTLE.

Record length. The length of each record of information returned. For variable length records, this value is set to 0. For variable length records, you can obtain the length of individual records from the records themselves.

Records returned. The number of records that are returned in the receiver variable. This is the smallest of the following values:

- The number of records that fit into the receiver variable.
- The number of records in the list.
- The number of records that was requested.

Request handle. The handle of the request that can be used for subsequent requests of information from the list. The handle is valid until the Close List (QGYCLST) API is called to close the list or until the job ends.

Note: This field should be treated as a hexadecimal field. It should not be converted from one CCSID to another, for example, EBCDIC to ASCII, because doing so could result in an unusable value.

Reserved. A reserved field. This field must be set to hexadecimal or binary zero.

Total records. The total number of records available in the list.

Error Messages

Message ID	Error Message Text
CPF1E65 E	Library backup list in use.
CPF1E67 D	Backup options and library backup list damaged.
CPF1E6B E	Folder backup list in use.
CPF1E6D D	Folder backup list damaged; new one created.
CPF1EC5 E	Backup option &1 is not valid.
CPF1EE4 E	Not authorized to run backup.
CPF1EEA E	Not authorized to library backup list.
CPF1EEB E	Not authorized to folder backup list.
CPF1E99 E	Unexpected error occurred.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C31 E	Object type &1 is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF8148 E	Object &4 type &3 in &9 damaged.
CPF8A77 E	Folder &1 not found.
CPF8A78 E	Folder &1 in use.
CPF8A79 E	Folder &1 is logically damaged.
CPF8A80 E	Document &2 in use in folder &1.
CPF9012 E	Start of document interchange session not successful for &1.
CPF9032 E	Document interchange session not started.
CPF9076 E	Internal system error processing documents or folders.
CPF9095 E	Folder &1 is damaged.
CPF909A E	Document &2 in folder &1 is damaged.

Message ID	Error Message Text
CPF9810 E	Library &1 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPF9830 E	Cannot assign library &1.
CPF9838 E	User profile storage limit exceeded.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V3R7

Restore from Application (QaneRsta) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	User space format name	Input	Char(8)
3	Status format name	Input	Char(8)
4	Status information	Output	Char(*)
5	Length of status information	Input	Binary(4)
6	Error code	I/O	Char(*)

Service Program Name: QANESERV
Default Public Authority: *USE
Threadsafe: No

The Restore from Application (QaneRsta) API enables an application to provide the restore records that are required for a restore-from-save-file operation. The application defines the restore operation by specifying the type of restore command, and by providing the restore command parameters. The API calls an exit program to retrieve the restore records from the application instead of from the save file.

To use the API, the application must provide the following:
- A user space that contains the required input parameter group
- An exit program

When processing the restore command, the API does the following:
- Calls the exit program to indicate the start of the transfer sequence
- Submits the restore command for processing
- Calls the exit program repeatedly to transfer the restore records
- Calls the exit program to signal the end of the restore operation
- May call the exit program to force an abnormal end to the restore operation

The program that calls the API is suspended while the restore operation is being processed.

Restrictions

QTEMP should not be specified for the library name on the OUTFILE parameter because the restore command is submitted by a prestart job running in the QSYSWRK subsystem and not in the job that called the API. Locks should not be applied to restore objects that would conflict with locks applied by the restore operation running in the prestart job.

Objects can be restored by this API only if the objects were saved using the "Save to Application (QaneSava) API" on page 196 (QaneSava) API, and only if the objects were saved from the current or an earlier release of the operating system.

The application must provide the restore records in the order presented, without modification, for the objects to be successfully restored.

≫ Because the restore command is processed from within a prestart job, the adopted authority of the thread using the Restore from Application (QaneRsta) API is not available to the restore command. One way to give the prestart job more authority is to use the adopted authority to swap user profiles within the application before calling the QaneRsta API. ≪

Authorities and Locks

Exit Program Library Authority
> *EXECUTE

Exit Program Authority
> *EXECUTE

User Space Lock
> *SHRNUP

User Space Library Authority
> *USE

User Space Authority
> *USE

Restore Command Library Authority
> *EXECUTE

Restore Command Authorities
> See the restore command

Restored Object Locks
> See the Back up your server topic.

Restored Object Authorities

> See Appendix D in the iSeries Security Reference book.

Required Parameter Group

Qualified user space name
> INPUT; CHAR(20)
>
> The user space that contains all the control information for the restore operation. The first 10 characters contain the user space name. The second 10 characters contain the name of the library where the user space is located.
>
> You can use the following special values for the library name:

CURLIB   The job's current library is used to locate the user space. If no library is specified as the current library for the job, the QGPL library is used.

LIBL   The actual library that is used is returned in the status information. the user space.

> The library list is used to locate the user space.

User space format name
> INPUT; CHAR(8)
>
> The format name for the input parameters that are contained in the user space. For the format of the structure, see "SVRS0100 Format" on page 43.

Status format name
> INPUT; CHAR(8)

The format name for the status information returned on the API call. For the format of the structure, see "SRST0100 Format" on page 45.

Status information
OUTPUT; CHAR(*)

The status information returned on the API call.

Length of status information
INPUT; BINARY(4)

The length of the status information returned on the API call. The minimum length is 8 bytes.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

SVRS0100 Format

This format defines the input parameter group for the API.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Length of structure
4	4	BINARY(4)	Offset to restore command parameters
8	8	BINARY(4)	Length of restore command parameters
12	C	BINARY(4)	Offset to application data
16	10	BINARY(4)	Length of application data
20	14	BINARY(4)	Restore command type
24	18	CHAR(10)	Exit program name
34	22	CHAR(10)	Exit program library
44	2C	CHAR(8)	Target release
		CHAR(*)	Restore command parameters
		CHAR(*)	Application data

Field Descriptions

Application data. Information that the application wants passed to the exit program. The content of this information is defined by the application. This field could contain information specific to the object being saved (such as the object name, size, and so forth), or it could contain the qualified name of another object that contains this information.

Exit program library. The name of the library that contains the exit program called by the API. the exit program.

Exit program name. The name of the exit program that is called by the API. See "Restore from Application Exit Program" on page 205 exit program for additional details.

Length of application data. The length of the application data. This value is passed to the exit program. This value must be set to zero if there is no application data.

Length of restore command parameters. The length of the restore command parameters. The maximum allowable length is 32500 bytes for restore commands.

Length of structure. The length of this structure, from the start of the input parameters to the last byte of the application data.

Offset to application data. The byte offset from the beginning of the user space to the start of the application data. This value must be set to zero if there is no application data.

Offset to restore command parameters. The byte offset from the beginning of the user space to the start of the restore command parameters.

» Restore command parameters. A character string that contains the restore command parameters or restore keys. These parameters are validated when the API submits the command for processing. Refer to the restore commands in the Control Language (CL) for detailed information about valid parameters. Refer to the "Restore Object (QsrRestore) API" on page 46 (QsrRestore) API or "Restore Object List (QSRRSTO) API" on page 60 (QSRRSTO) API for detailed information about valid keys. «

These additional restrictions apply to the restore command parameters when you use this API:
- The parameters must be consistent with the restore command type.
- The parameters must not include the restore command name.
- The parameters must be separated by at least one blank character.
- » The Device (DEV) and Save file (SAVF) parameters must not be used. These parameters are provided by the API.«
- » The End of media option (ENDOPT), Label (LABEL), Media definition (MEDDFN), Optical file (OPTFILE), Sequence number (SEQNBR) and Volume identifier (VOL) parameters must not be used. These parameters are inconsistent with the save file operation provided by the API.«
- Only single library names can be used with the SAVLIB or RSTLIB parameter.

The following examples illustrate the restore command parameters that are required for typical restore scenarios:
- Example 1: Restore command type 1 (RST)

  ```
  OBJ('/*') ('/QSYS.LIB' *OMIT)
  ('/QDLS.LIB' *OMIT)
  ```

  These parameters restore all objects that are not in libraries and that are not document library objects.
- Example 2: Restore command type 2 (RSTOBJ)

  ```
  OBJ(FILE*) SAVLIB(MYLIB) OBJTYPE(*FILE)
  SAVDATE(122297) RSTLIB(TMPLIB)
  ```

  These parameters restore all files with names that start with the characters FILE* to library TMPLIB that were saved from the library named MYLIB on 22 December 1997.
- Example 3: Restore command type 4 (RSTLIB)

  ```
  SAVLIB(JOE)
  ```

  These parameters restore the library named JOE.

» These additional restrictions apply to the command parameters when you use the Restore Object (QsrRestore) API or Restore Object List (QSRRSTO) API. «
- The keys specified must be consistent with restore from save file operations.
- » The Device (DEV) and Save file (SAVF) keys must not be used. These keys are provided by this API. «
- » The End of media option (ENDOPT), Label (LABEL), Media definition (MEDDFN), Optical file (OPTFILE), Sequence number (SEQNBR) and Volume identifier (VOL) keys must not be used. These keys are inconsistent with the save file operation provided by the API.«

- The starting offset for the keys is always 0 and not the offset of the restore command parameters.
- All integer values within the keys must be aligned on 4-byte boundaries.
- All pointers within the keys must be aligned on 16-byte boundaries.

Restore command type. The type of restore command that is to be processed.

1	Restore (RST) command
2	Restore Object (RSTOBJ) command
3	Restore Document Library Object (RSTDLO) command
4	Restore Library (RSTLIB) command
5	Restore Object (QsrRestore) API
≫*6*	Restore Object List (QSRRSTO) API
7	Restore System Information (RSTSYSINF) command ≪

Target release.

An ignored field. Must be set to blanks.

SRST0100 Format

This format defines the status information that is returned on the API call.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Transfer time
12	C	BINARY(4)	Transfer block size
16	10	BINARY(4)	Transfer block multiplier
20	14	BINARY(4)	Last block size
24	18	CHAR(10)	User space library used
≫ 34	22	CHAR(2)	Reserved ≪
≫ 36	24	BINARY(4)	Decimal transfer time ≪

Field Descriptions

Bytes returned. The number of status information bytes returned. If the value specified in the length of status information parameter is larger than the specified status information structure, this value is set to the last byte of the returned information.

Bytes available. The number of status information bytes available for the specified status information format.

≫ **Decimal transfer time.** The decimal portion of the transfer time in millionths of a second. If the value returned for Transfer time was 5 and the value returned for Decimal transfer time was 827352, then the total transfer time was 5.827352 seconds. ≪

≫ **Reserved.** This field is unused and will be returned with blanks. ≪

Transfer block size. The number of bytes in the blocks transferred by the exit program.

Transfer block multiplier. The number of blocks successfully transferred by the exit program.

Last block size. The number of bytes in the last block transferred by the exit program.

The true transfer size of the operation is equal to the transfer block size multiplied by the transfer block multiplier plus the last block size.

Transfer time. The elapsed time, in seconds, that begins when the application calls the API, and ends when the API returns to the caller.

User space library used. The name of the user space library that is used in the API call.

Error Messages

Message ID	Error Message Text
CPF3700 E	All CPF37xx messages could be signalled. xx is from 01 to FF, excluding tape and diskette errors since the operation is a restore from a save file.
CPF3800 E	All CPF38xx messages could be signalled. xx is from 01 to FF, excluding tape and diskette errors since the operation is a restore from a save file.
CPF2115 E	Object &1 in &2 type *&3 damaged.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF8100 E	All CPF81xx messages could be returned. xx is from 01 to FF.
CPF9800 E	All CPF98xx messages could be signaled. xx is from 01 to FF.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.
CPFB8C0 E	Status information length for &1 API is not valid.
CPFB8C1 E	Unsupported value for &1 API.
CPFB8C2 E	Offset value for &1 API not valid. Reason &6.
CPFB8C3 E	Length value for &1 API not valid. Reason &6.
CPFB8C4 E	Unexpected condition with exit program for &1 API. Reason &6.
CPFB8C5 E	Parameter &2 specified more than once for API &1.
CPIB8C6 I	Trace data generated for API &1.
CPFB8C7 E	Unsupported value for &1 API.
CPFB8C8 E	Command syntax error detected by &1 API.
CPFB8C9 E	Command exception occurred using &1 API.
CPFB8CF E	Unexpected condition with &1 API. Reason &6.

API introduced: V4R3

Restore Object (QsrRestore) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Error code	I/O	Char(*)

Service Program Name: QSRLIB01
Default Public Authority: *USE
Threadsafe: No

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

Authorities and Locks

User Space

> *User Space Authority*
> > *USE
>
> *User Space Library Authority*
> > *EXECUTE
>
> *User Space Lock*
> > *EXCLRD

Objects to Be Restored, ≫ Created Parent Directories, ≪ Devices, and Output

> *Locking*
>
> > See the Backup and Recovery book for information on object locking for the Restore Object (RST) command.
>
> *Authority*
>
> > In the iSeries Security Reference book, see the Appendix about authorities required for the Restore Object (RST) command.

Required Parameter Group

Qualified user space name
> INPUT; CHAR(20)
>
> The user space that is to hold all the information for the restore operation. The first 10 characters contain the user space name. The second 10 characters contain the name of the library where the user space is located. See "User Space Format" for the format of the information in the user space.
>
> You can use the following special values for the library name. It should be noted, however, that the library name that is actually used is not passed back to the user. Care should be taken when you use these special values to avoid unexpected results.

CURLIB	The job's current library is used to locate the user space. If no library is specified as the current library for the job, the QGPL library is used.
LIBL	The library list is used to locate the user space.

Error code
> I/O; CHAR(*)
>
> The structure in which to return error information. For the format of the structure, see Error Code Parameter

User Space Format

The following defines the format for the information in the user space. For detailed descriptions of the fields in the user space format, see "Field Descriptions" on page 48.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Number of variable length records
4	4	BINARY(4)	Offset to first record

Offset			
Dec	Hex	Type	Field
8	8	CHAR(8)	Reserved
Note: These fields repeat for each variable length record.			
		BINARY(4)	Key
		BINARY(4)	Offset to next record
		CHAR(8)	Reserved
		CHAR(*)	Data

If the length of the data is longer than the key identifier's data length, the data will be truncated at the right. No message will be issued.

If the specified data length is shorter than the key field's defined data length, an error message is returned for binary fields. If the field is a character field, the data is padded with blanks and an error message will not be returned.

Note: This does not apply to keys that allow a list of values to be specified. In these cases, the amount of data read is based on the specified number of entries in the list.

If keys are duplicated in the user space, only the last value for a given key is used for the restore operation.

Each variable length record must be 4-byte aligned. If not, unpredictable results may occur.

Field Descriptions

Data. The data used to specify the value for the given key.

Key. The parameter of the Restore Object (RST) command to specify. See "Valid Keys" for the list of valid keys.

Offset to first record. The offset from the beginning of the user space to the first variable length record.

Offset to next record. The offset from the beginning of the user space to the next variable length record.

Number of variable length records. The number of variable length records that are passed in the user space. ≫The valid range is from 2 through 19. ≪

Reserved. This field should contain x'00's.

Valid Keys

The following table lists the valid keys for the key field area of the variable length record. For detailed descriptions of the keys, see the "Field Descriptions" on page 49.

Some messages for this API refer to parameters and values of the Restore Object (RST) command. This table can also be used to locate the key names that correspond to the RST command parameters. The field descriptions contain, in addition to detailed descriptions, the corresponding parameter values.

The object path name key and the device path name key are required keys. The other keys are optional.

Key	Type	Field	RST Command Parameter
1	CHAR(*)	Device path name	DEV
2	CHAR(*)	Object path name	OBJ
3	CHAR(1)	Directory subtree	SUBTREE
4	CHAR(1)	System	SYSTEM
5	CHAR(7)	Save date	SAVDATE
6	CHAR(8)	Save time	SAVTIME
7	CHAR(1)	Option	OPTION
8	CHAR(*)	Allow object differences	ALWOBJDIF
9	CHAR(2)	Force object conversion	FRCOBJCVN
10	CHAR(*)	Volume identifier	VOL
11	CHAR(*)	Label	LABEL
12	BINARY(4)	Sequence number	SEQNBR
13	CHAR(1)	End of media option	ENDOPT
14	CHAR(*)	Optical file	OPTFILE
15	CHAR(*)	Output	OUTPUT, INFTYPE
16	CHAR(*)	Object ID	OBJID
17	CHAR(*)	Name pattern	PATTERN
≫ 18	CHAR(1)	Create parent directories	CRTPRNDIR ≪
≫ 19	CHAR(10)	Parent directory owner	PRNDIROWN ≪

Field Descriptions

The values shown in parentheses are the corresponding values for the RST command parameters.

Allow object differences. Whether differences are allowed between the saved object and the restored object. The differences include:

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
Note: This field repeats for each allow object difference value.			
		CHAR(1)	Allow object difference

Number in array. The number of allow object difference values. The possible values are:

1-3   The number of allow object difference values.

Allow object difference. Whether differences are allowed between the saved object and the restored object. The differences include:

≫

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

  Note: This key has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.

- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.

- Primary group: The primary group of an object on the system is different from the primary group of an object from the save operation.

«

The default is 0. The possible values are:

0 No differences are allowed between the saved object and the restored object. If 0 is specified for the allow object difference field, 1 must be specified for the number in array field. (*NONE)

»

If an object already exists on the system with a different owner or primary group than the saved object, the object is not restored.

If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored.

If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

«

1 All differences are allowed between the saved object and the restored object. If 1 is specified for the allow object difference field, 1 must be specified for the number in array field. (*ALL)

»

If an object already exists on the system with a different owner or primary group than the saved object, the object is restored with the existing values.

If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object.

If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority of the object is set to *EXCLUDE.

«

2 » Authorization list differences are allowed. (*AUTL)

If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object.

If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority of the object is set to *EXCLUDE.

«

3 Ownership differences are allowed. (*OWNER)

  » If an object already exists on the system with a different owner than the saved object, the object is restored with the existing value. «

4 Primary group differences are allowed. (*PGP)

  » If an object already exists on the system with a different primary group than the saved object, the object is restored with the existing value. «

»

Create parent directories. Whether parent directories of objects being restored should be created if they do not exist. For example, if object ′/a/b/c/file1′ is being restored then directories ′/a′, ′/a/b′ and ′/a/b/c′ must exist. This key only applies to ″root″ (/), QOpenSys and user-defined file systems, it will be ignored for all other file systems. The default is 0. The possible values are:

0 Parent directories will not be created if they do not exist. Diagnostic message CPD375B will be sent and the object will not be restored. (*NO)

1 The restore will create parent directories if they do not exist. The directories created by the restore will have *EXCLUDE public authority and will be owned by the user profile specified using key 19 - *Parent directory owner*. (*YES)

Note: Other than owner, the parent directory attributes will be set as though the Create Directory (CRTDIR) command had been used to create the parent directory without changing any of the command defaults (the defaults are with respect to the defaults on the shipped command and do not reflect changes that have been made to the command default values). «

Device path name. The path name of the device from which the objects are restored.

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Number in array
		BINARY(4)	Offset to first device path name
Note: These fields repeat for each device name.			
		BINARY(4)	Offset to next device path name
		CHAR(12)	Reserved
		CHAR(*)	Device path name

Device path name. The path name of the device used for the restore operation. The path name should be specified in the Qlg_Path_Name_T format. If a pointer is specified in the path name format, it must be 16-byte aligned. If not, unpredictable results may occur. For more information on this structure, see Path name format. The possible value is:

device-path-name The path name of the » media definition, « media library device, optical device, save file, or tape device used to restore the objects. If a » media definition, « media library device, optical device, or save file path name is specified, it must be the only element in the array.

  » For information about creating and using a media definition, see the Backup and Recovery book and the "Create Media Definition (QSRCRTMD, QsrCreateMediaDefinition) API" on page 14 (QSRCRTMD, QsrCreateMediaDefinition) API. «

Number in array. The number of devices used during the restore operation. The possible value is:

1-4 The number of devices used during the restore operation.

Offset to first device path name. The offset from the beginning of the user space to the first device path name in the list. The possible value is:

n The offset from the beginning of the user space to the first device path name in the list.

Offset to next device path name. The offset from the beginning of the user space to the next device path name in the list. The possible value is:

n The offset from the beginning of the user space to the next device path name in the list. If the current device path name is the last device path name in the array, this value should be 0.

Reserved. Reserved. The possible value is:

x'00' This field should contain x'00's.

Directory subtree. Whether the directory subtrees are included in the restore operation. The default is 1. The possible values are:

0 No subtrees are included in the restore operation. If a directory matches the object name pattern specified, the objects in the directory are included. If the directory has subdirectories, neither the subdirectories nor the objects in the subdirectories are included. (*NONE)
1 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. (*ALL)
2 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. (*DIR)
3 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. (*OBJ)
4 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). (*STG)

End of media option. The operation that is performed automatically on the tape or optical volume after the restore operation ends. If more than one volume is used, this key applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached. The default is 0.

Note: This parameter is valid only if a tape or optical device name is specified. For optical devices, 2 is the only value supported; 0 and 1 are ignored.

The possible values are:

0 The tape is automatically rewound, but not unloaded, after the operation ends. (*REWIND)
1 The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive. (*LEAVE)
2 The tape is automatically rewound and unloaded after the operation ends. Some optical devices eject the volume after the operation ends. (*UNLOAD)

Force object conversion. Whether to convert user objects to the format required for use in the current version of the operating system.

Note: If an object needs to be converted, but is not converted during the restore operation, the object is automatically converted the first time it is used.

Offset			
Dec	Hex	Type	Field
		CHAR(1)	Force conversion
		CHAR(1)	Objects to convert

Force conversion. Whether objects should be converted on the restore operation. The default is 2. The possible values are:

0   The objects are not converted during the restore operation. (*NO)

1   The objects are converted during the restore operation. (*YES)

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

2   The objects are converted based on the value of the QFRCCVNRST system value. (*SYSVAL)

> **Note:** If this value is specified and the system value QFRCCVNRST has a value of 1, the restore operation proceeds as if 1 were specified for the force conversion field and 2 were specified for the objects to convert field.
>
> If QFRCCVNRST has a value of 0, the restore operation proceeds as if 0 were specified for the force conversion field.

Objects to convert. Which objects should be converted on the restore operation. The default is 2. The possible values are:

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

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

Object ID. Whether the object ID of the restored object will be the object ID of the object from the save media, the object ID of the object that exists on the system prior to the restore, or a new object ID generated by the system if the object does not exist on the system prior to the restore. The default is 0. The possible values are:

0   The restored object will have the object ID it had when it was saved. (*SAVED)

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

Label. The file identifier of the media to be used for the restore operation. The default is *SEARCH. The possible values are:

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

file-identifier   The identifier (maximum of 17 characters) of the tape file used for the restore operation.

Name pattern. Specifies a pattern to be used to include or omit objects.

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array

Offset		Type	Field
Dec	Hex		
		BINARY(4)	Offset to first pattern name
Note: These fields repeat for each pattern name.			
		BINARY(4)	Offset to next pattern name
		CHAR(1)	Option
		CHAR(11)	Reserved
		CHAR(*)	Pattern name

Number in array. The number of pattern names. The possible value is:

1-n   The number of pattern names.

Pattern name. Specifies a pattern name. The possible value is:

pattern-name   The object name or pattern that can match many names.

Offset to first pattern name. The offset from the beginning of the user space to the first pattern name. The possible value is:

n   The offset from the beginning of the user space to the first pattern name.

Offset to next pattern name. The offset from the beginning of the user space to the next pattern name in the list. The possible value is:

n   The offset from the beginning of the user space to the next pattern name in the list. If the current pattern name is the last pattern in the array, this value should be 0.

Option. Whether names that match the pattern should be included or omitted from the restore operation.

Note: The subtree key specifies whether the subtrees are included or omitted.

The possible values are:

0   All objects which are included by the OBJ parameter are included in the restore except those objects which match the PATTERN parameter. This value overrides objects that are included with option 1 and is intended to be used to omit a subset of a previously selected patterns. (*OMIT)
1   Only objects which are included by the OBJ parameter and match the PATTERN parameter are included in the restore, unless overridden by an omit specification. (*INCLUDE)

Reserved Reserved. The possible value is:

x'00'   This field should contain x'00's.

Object path name. The path name of the object to restore. You can specify a pattern for this path name.

Offset		Type	Field
Dec	Hex		
		BINARY(4)	Number in array
		BINARY(4)	Offset to first object path name
Note: These fields repeat for each object path name.			
		BINARY(4)	Offset to next object
		BINARY(4)	Offset to new object path name
		CHAR(1)	Option
		CHAR(7)	Reserved
		CHAR(*)	Object path name
		CHAR(*)	New path name

New path name. The new path name of the object. The path name should be specified in the Qlg_Path_Name_T format. If a pointer is specified in the path name format, it must be 16-byte aligned. If not, unpredictable results may occur. For more information on this structure, see Path name format. The possible value is:

path-name   The path name with which to restore the object. If a pattern is specified for the object path name field, the new path name must be an existing directory in which to restore any objects that match the pattern. If an object name is specified in the object path name field, each component in the new path name must exist with the exception of the last component. If the object described in the last component does not exist, it will be restored as new.

Number in array. The number of object path names to be restored. The possible value is:

1-300   The number of object path names to be restored.

Object path name. The path names of the objects saved on the media. Directory abbreviations (for example, the current directory) are expanded with their current values, not with the values they had at the time of the save operation. The path name should be specified in the Qlg_Path_Name_T format. If a pointer is specified in the path name format, it must be 16-byte aligned. If not, unpredictable results may occur. For more information on this structure, see Path name format. The possible value is:

path-name   The path name or pattern that can match many names. If 0 is specified for the new name option, each component in the path name must exist with the exception of the last component. The name in the last component is restored as new if it does not exist.

Offset to first object path name. The offset from the beginning of the user space to the first object path name. The possible value is:

n   The offset from the beginning of the user space to the first object path name.

Offset to new object path name. The offset from the beginning of the user space to the new object path name. The possible values are:

0   There is not a new object path name. The object will be restored to the same name with which it was saved.
n   The offset from the beginning of the user space to the new object path name.

Offset to next object. The offset from the beginning of the user space to the next object in the list. The possible value is:

n   The offset from the beginning of the user space to the next object in the list. If the current object is the last object in the array, this value should be 0.

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

Note: The subtree key specifies whether the subtrees are included or omitted.

The possible values are:

0   The objects that match the object name pattern are not restored. This value overrides the objects that are included with option 1 and is intended to be used to omit a subset of a previously selected pattern. (*OMIT)
1   The objects that match the object name pattern are restored, unless overridden by an omit specification. (*INCLUDE)

Reserved Reserved. The possible value is:

x'00'   This field should contain x'00's.

Optical file. The path name of the optical file that is used for the restore operation. The path name should be specified in the Qlg_Path_Name_T format. If a pointer is specified in the path name format, it must be 16-byte aligned. If not, unpredictable results may occur. For more information on this structure, see Path name format. The default is '*'. The possible values are:

''*	The system searches the root directory of the optical volume for the default name generated by the corresponding save operation.
'Optical-directory-path-name/'*	The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.
Optical file path name	The path name of the optical file that is used for the restore operation, beginning with the root directory of the volume.

Option. Whether to restore objects that already exist on the system or to restore objects that do not already exist on the system. The default is 0. The possible values are:

0   All of the specified objects are restored, whether or not they already exist on the system. (*ALL)
1   Objects are restored only if they do not already exist on the system. (*NEW)
2   Objects are restored only if they already exist on the system. (*OLD)

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

Offset			
Dec	**Hex**	**Type**	**Field**
		CHAR(1)	Option
		CHAR(1)	Type of output information
		CHAR(14)	Reserved
		CHAR(*)	Output path name

Option. Whether a list of information about the restored objects is created. The default is 0. The possible values are:

0   No output is created. (*NONE)
1   The output is printed with the job's spooled output. (*PRINT)
2   The output is directed to an existing stream file or user space specified by the output path name.

Output path name. The path name should be specified in the Qlg_Path_Name_T format. If a pointer is specified in the path name format, it must be 16-byte aligned. If not, unpredictable results may occur. For more information on this structure, Path name format. The possible value is:

path-name   The path name of the existing stream file or user space to which the output of the command is directed.

Reserved Reserved. The possible value is:

x'00'   This field should contain x'00's.

Type of output information. The type of information that should be directed to the spooled file, stream file, or user space specified for the output key. The possible values are:

0   The file will contain information about the command, and an entry for each directory. (*SUMMARY)
1   The file will contain information about the command, an entry for each directory, and an entry for each object that was not successfully restored. (*ERR)
2   The file will contain information about the command, an entry for each directory, an entry for each object that was successfully restored, and an entry for each object that was not successfully restored. (*ALL)

≫

Parent directory owner. The name of an existing user profile that will own parent directories that are created by the restore. This key only applies to "root" (/), QOpenSys and user-defined file systems, it will be ignored for all other file systems.. The default is *PARENT. The possible values are:

*PARENT   The owner of the parent directory being created by the restore will be copied from the directory it is being created into. (*PARENT)
name   Specify a user profile to be the owner of any parent directories that are created by the restore. (name) ≪

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

date   The date the objects were saved, in the format CYYMMDD:

      C   Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.

      YY   Year

      MM   Month

      DD   Day

Save time. The time the objects were saved. If this key is not specified, the version of the objects to be restored is the first version on the volume.

Note:

1. This key is valid only if the save date key is specified.

2. This key is ignored when the sequence number key is specified.

The possible value is:

time   The time the objects were saved in the format HHMMSS:

        HH   Hour

        MM   Minute

        SS   Second

Sequence number. The tape file sequence number to be used. The default is -1. The possible values are:

-1	The tape volume is searched for the next file that contains any of the specified objects. (*SEARCH)
1-16777215	The sequence number of the file.

System. Whether to process objects that exist on the local system or remote systems. The default is 0. The possible values are:

0	Only local objects are processed. (*LCL)
1	Only remote objects are processed. (*RMT)
2	Both local and remote objects are processed. (*ALL)

Volume identifier. The volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, from which data is to be restored. The volumes must be placed in the device in the order specified on this key. After all specified volumes are filled, the restore operation continues on whatever volumes are mounted on the device.

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
		BINARY(4)	Length of each volume identifier
		BINARY(4)	Offset to first volume identifier
Note: These fields repeat for each volume identifier			
		BINARY(4)	Offset to next volume identifier
		CHAR(*)	Volume identifier

Length of each volume identifier. The character length of each of the volume identifiers. The possible value follows:

n   The size of a single volume identifier. The maximum size of a tape volume identifier is 6 characters. The maximum size of an optical volume identifier is 32 characters. If a volume identifier larger than the maximum size is entered for this key, it is truncated to the maximum size.

Number in array. The number of volume identifiers that are used during the restore operation. The default is 0. The possible values are:

0 The volume currently placed in the device is used. If 0 is specified for a tape media library device, volume identifiers must be supplied by using the Tape Management exit program during the save or restore operation. If 0 is specified, the length of each volume identifier value is ignored. (*MOUNTED)

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

1-75 The number of volume identifiers used during the restore operation.

Offset to first volume identifier The offset from the beginning of the user space to the first volume identifier in the list. The possible value is:

n The offset from the beginning of the user space to the first volume identifier in the list.

Offset to next volume identifier The offset from the beginning of the user space to the next object volume identifier in the list. The possible value is:

n The offset from the beginning of the user space to the next volume identifier in the list. If the current volume identifier is the last volume identifier in the array, this value should be 0.

Volume identifier. The volume identifiers of one or more volumes to be used. The possible value is:

Volume identifier The volume identifiers of one or more volumes to be used.

Dependencies between Keys

The following two tables list the dependencies between the different keys. If the dependency pertains only to a certain value, then that value is also shown (key = *n*, where *n* is the value). Otherwise, if the dependency is true for all values of the key, then only the name of the key is given.

The following table lists the conditions where specifying a certain key forces the use of another key.

If you specify...	...must be specified
Device = optical library device	Volume identifier

The following table lists the conditions where specifying a certain key excludes the user from using another key or a particular value of that key.

If you specify...	...cannot be specified
≫Create parent directories = 0	Parent directory owner ≪
≫Device = media definition	Optical file Sequence number Volume identifier ≪
Device = optical device	Label Sequence number
Device = save file	End of media option Label Optical file Sequence number Volume identifier
Device = tape device	Optical file

Relationship to RST Command

Because of the relationship between the QsrRestore API and the RST command, the following situations should be noted:

- Message text: Several messages produced by this API refer to parameters or values of the RST command (for example, *SEARCH). To determine which key a given parameter corresponds to, see "Valid Keys" on page 48. To determine which key value a given parameter value corresponds to, see "Field Descriptions" on page 49.

- Command type: The command type listed for the API on headings of displays and print files is QsrRestore for integrated file system objects. If QsrRestore is used to restore objects in libraries or document library objects (DLOs), the generated output indicates that the corresponding library or DLO command generated the media.

Error Messages

Message ID	Error Message Text
CPF0001 E	Error found on &1 command.
CPF24B4 E	Severe error while addressing parameter list.
CPF3700 E	All CPF37xx messages could be signalled. xx is from 01 to FF.
CPF3800 E	All CPF38xx messages could be signalled. xx is from 01 to FF.
CPF3C31 E	Object type &1 is not valid.
CPF3C4D E	Length &1 for key &2 not valid.
CPF3C81 E	Value for key &1 not valid.
CPF3C82 E	Key &1 not valid for API &2.
CPF3C83 E	Key &1 not allowed with value specified for key &2.
CPF3C84 E	Key &1 required with value specified for key &2.
CPF3C85 E	Value for key &1 not allowed with value for key &2.
CPF3C86 E	Required key &1 not specified.
CPF3C87 E	Key &1 allows one value with special value.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF5729 E	Not able to allocate object &1.
CPF9800 E	All CPF98xx messages could be signaled. xx is from 01 to FF.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V4R3

Restore Object List (QSRRSTO) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Error Code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Restore Object List (QSRRSTO) API restores a list of objects or spooled files specified by the user. The list of objects, as well as any additional information needed for the restore operation, is specified by the user in a user space.

Authorities and Locks

User Space

> *User Space Authority*
>> *USE
>
> *User Space Library Authority*
>> *EXECUTE
>
> *User Space Lock*
>> *SHRNUP

Objects to Be Restored

The following authorities are needed if the user does not have save system (*SAVSYS) special authority. To allow any object differences, you need *ALLOBJ special authority.

> *Object Authority*
>> *OBJEXIST
>
> *Library Authority*
>> *EXECUTE, *READ, and *ADD
>
> *Object Lock*
>> *EXCL
>
> *Library Lock*
>> *SHRUPD

Spooled Files to Be Restored

If the user has save system (*SAVSYS) special authority, the following authorities are not needed.

> *Output Queue Authority*
>> *OBJEXIST
>
> *Output Queue Library Authority*
>> *EXECUTE
>
> *Output Queue Lock*
>> *EXCLRD

Note: Additional authority may be needed to change spooled file attributes. See "New Attributes Format" on page 79 for more information.

Devices

> *Save File Authority*
>> *USE
>
> *Save File Library Authority*
>> *EXECUTE
>
> *Save File Lock*
>> *EXCLRD
>
> *Tape or OpticalAuthority*
>> *USE
>
> *Tape or OpticalLock*
>> *EXCL

> *Media Library Device Lock*
>> *SHRUPD
>
> *Media Definition Authority*
>> *USE
>
> *Media Definition Library Authority*
>> *EXECUTE
>
> *Media Definition Lock*
>> *EXCLRD
>
> *Auxiliary Storage Pool (ASP)*
>> *USE

Output Files

> *Output File Lock*
>> *SHRRD

If the output file does not exist:

> *Output File Library Authority*
>> *READ and *ADD

If the output file exists and a new member will be added:

> *Output File Authority*
>> *OBJMGT, *OBJOPR, and *ADD
>
> *Output File Library Authority*
>> *EXECUTE and *ADD

If the output file exists and an existing member will be appended:

> *Output File Authority*
>> *OBJMGT and *ADD
>
> *Output File Library Authority*
>> *EXECUTE

If the output file exists and an existing member will be replaced:

> *Output File Authority*
>> *OBJMGT, *OBJOPR, *ADD, and *DLT
>
> *Output File Library Authority*
>> *EXECUTE

Required Parameter Group

Qualified user space name
> INPUT; CHAR(20)

The user space that is to hold all the information for the restore operation. The first 10 characters contain the user space name. The second 10 characters contain the name of the library where the user space is located. See "User Space Format" for the format of the information in the user space.

You can use the following special values for the library name. It should be noted, however, that the library name that is actually used is not passed back to the user. Care should be taken when using these special values to avoid unexpected results.

*CURLIB   The job's current library is used to locate the user space. If no library is specified as the current library for the job, the QGPL library is used.

*LIBL   The library list is used to locate the user space.

Error code
> I/O; CHAR(*)
>
> The structure in which to return error information. For the format of the structure, see Error Code Parameter.

User Space Format

The following defines the format for the information in the user space. For detailed descriptions of the fields in the user space format, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Number of variable length records
Note: These fields repeat for each variable length record.			
		BINARY(4)	Length of variable length record
		BINARY(4)	Key
		BINARY(4)	Length of data
		CHAR(*)	Data

If you specify a data length that is longer than the key field's defined data length, the data is truncated at the right. No error message is returned.

If you specify a data length that is shorter than the key field's defined data length, an error message is returned for binary fields. If the field is a character field, the data is padded with blanks.

Note: This does not apply to keys that allow a list of values to be specified. In these cases, the amount of data read is based on the specified number of entries in the list.

If keys are duplicated in the user space, only the last value for a given key is used for the restore operation.

It is recommended, but not required, to align each variable length record on a 4-byte boundary. That is, you should make the length of each variable length record a multiple of 4, even if the data length is not a multiple of 4.

Field Descriptions

Data. The data used to specify the value for the given key.

Key. The parameter of the Restore Object (RSTOBJ) command to specify. See "Valid Keys" on page 64 for the list of valid keys.

Length of data. The length of the data used to specify the value for the given parameter.

Length of variable length record. The length of the variable length record.

Number of variable length records. The number of variable length records that are passed in the user space. The valid range is from 2 through 27.

Valid Keys

The following table lists the valid keys for the key field area of the variable length record. For detailed descriptions of the keys, see "Field Descriptions" on page 65.

Some messages for this API refer to parameters and values of the Restore Object (RSTOBJ) command. This table can also be used to locate the key names that correspond to the RSTOBJ command parameters. The field descriptions contain, in addition to detailed descriptions, the corresponding parameter values.

The library key and the device key are required keys. The other keys are optional.

Key	Type	Field	RSTOBJ Command Parameter
1	CHAR(*)	Object information	OBJ, OBJTYPE
2	CHAR(*)	Saved library	SAVLIB
3	CHAR(*)	Device	DEV
4	CHAR(20)	Save file	SAVF
6	CHAR(*)	Volume identifier	VOL
7	BINARY(4)	Sequence number	SEQNBR
8	CHAR(*)	Label	LABEL
10	CHAR(1)	End of media option	ENDOPT
17	CHAR(*)	File member	FILEMBR
23	CHAR(1)	Output	OUTPUT
24	CHAR(20)	Qualified output file	OUTFILE
25	CHAR(11)	Output member	OUTMBR
26	CHAR(1)	Type of output information	INFTYPE
27	CHAR(*)	Optical file	OPTFILE
29	CHAR(*)	Omit libraries	OMITLIB
30	CHAR(*)	Omit object information	OMITOBJ
31	CHAR(20)	Media definition	MEDDFN
35	CHAR(*)	Spooled file data	SPLFDTA
36	CHAR(1)	Option	OPTION
37	CHAR(1)	Database member option	MBROPT
38	CHAR(7)	Save date	SAVDATE
39	CHAR(6)	Save time	SAVTIME
40	CHAR(*)	Allow object differences	ALWOBJDIF
41	CHAR(2)	Force object conversion	FRCOBJCVN
42	CHAR(10)	Restore to library	RSTLIB
43	CHAR(10)	Restore to ASP device	RSTASPDEV
44	BINARY(4)	Restore to ASP number	RSTASP

Field Descriptions

The values shown in parentheses are the corresponding values for the RSTOBJ command parameters.

Allow object differences. Whether differences are allowed between the saved objects and the restored objects. The default is a single value of 0. For the format of this key, see "Allow Object Differences Key Format" on page 68.

Database member option. Which members are restored for database files that exist on the system. The default is 4. The possible values are:

1   All the members in the saved file are restored. (*ALL)

2   Only the members in the saved file that do not exist in the current version of the file on the system are restored. (*NEW)

3   Only the members in the saved file that do exist in the current version of the file on the system are restored. (*OLD)

4   The saved members are restored if the list of the members where they exist match, member for member, the lists of the current system version. (*MATCH)

Device. The names of the devices used for the restore operation. The device must already be known on the system by a device description. For the format of this key, see "Device Key Format" on page 70.

End of media option. The operation that is performed automatically on the tape or optical volume after the restore operation ends. If more than one volume is used, this key applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached. The default is 0.

Note: This parameter is valid only if a tape or optical device name is specified. For optical devices, 2 is the only value supported; 0 and 1 are ignored.

The possible values are:

0   The tape is automatically rewound, but not unloaded, after the operation ends. (*REWIND)

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

2   The tape is automatically rewound and unloaded after the operation ends. Some optical devices eject the volume after the operation ends. (*UNLOAD)

File member. A list of the database files and their members that are to be restored. Each database file specified here must also be specified in the list of objects to be restored. If this key is not specified, the default of *ALL will be used for both the file name and the member name. For the format of this key, see "File Member Key Format" on page 71.

Force object conversion. Whether to convert user objects to the format required for use in the current version of the operating system when the objects are restored. The default is 2 (*SYSVAL). For the format of this key, see "Force Object Conversion Key Format" on page 72.

Notes:

1. This key applies only to user objects of the *MODULE, *PGM, *SRVPGM, and *SQLPKG object types.

2. An object must have creation data (either observable or unobservable) to be converted.

3. If an object needs to be converted, but is not converted during the restore operation, the object is automatically converted the first time it is used.

Label. The name that identifies the data file on the tape. Although the label key is defined as CHAR(*), the maximum length of a label is currently 17. If the length of data field is specified as more than 17, the label is truncated such that only the first 17 characters are used. The default is *SAVLIB.

| *SAVLIB | The file label is the name of the library specified for the saved library key. |
| Data file identifier | The data file identifier of the data file used. This option is valid only for a single-library restore operation. |

Media definition. The name and library of the media definition that identifies the devices and media used to contain the restored data. For information about creating and using a media definition, see the

Backup and Recovery ![book icon] book and the "Create Media Definition (QSRCRTMD, QsrCreateMediaDefinition) API" on page 14 (QSRCRTMD, QsrCreateMediaDefinition) API. The first 10 characters contain the media definition name; the second 10 characters contain the library in which the media definition is located.

You can use these special values for the library name:

| *CURLIB | The job's current library is used to locate the media definition. If no library is specified as the current library for the job, the QGPL library is used. |
| *LIBL | The library list. |

Object information. A list of the name and type of the objects to be restored. If *ALL is specified for the object name and object type, the list cannot contain other entries. The default for both the object name and the object type is *ALL. For the format of this key, see "Object Information Key Format" on page 72.

Omit libraries. A list of the libraries to be omitted from the restore operation. The default is *NONE. For the format of this key, see "Omit Library Key Format" on page 73.

Omit object information. A list of the name and type of the objects and library to be omitted from the restore operation. If *ALL is specified for the object name and object type, the list cannot contain other entries. The default for both the object name and the object type is *ALL. For the format of this key, see "Omit Object Information Key Format" on page 73.

Optical file. The name that identifies the file on the optical volume. Although the optical file is defined as CHAR(*), the maximum length of an optical file name is currently 256 characters. If the length of data field is specified as more than 256 characters, the name is truncated such that only the first 256 characters are used. The default is '*'. The possible values are:

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

Option. Which objects are restored. The default is 1. The possible values are:

1	All the objects in the saved library are restored. (*ALL)
2	Only the objects in the saved library that do not exist in the current version of the library on the system are restored. (*NEW)
3	Only the objects in the saved library that do exist in the current version of the library on the system are restored. (*OLD)
4	Only the objects in the saved library that do exist in the current version of the library on the system with their storage freed are restored. (*FREE)

Output. Whether a list of information about the restored objects is created. The default is 0. The possible values are:

0	No output listing is created. (*NONE)
1	The output is printed with the job's spooled output. (*PRINT)
2	The output is directed to the database file specified with the output file key. (*OUTFILE)

Output member. The name of the database file member used to save the object information. This field also determines whether to replace or add the data if the member already exists. The defaults are *FIRST for the output member name field and 0 for the option field. For the format of this key, see "Output Member Key Format" on page 74.

Qualified output file. The qualified name of the database file to which the information about the objects is directed. This key is required only if the output key is set to 2. The first 10 characters contain the output file name; the second 10 characters contain the output file library. The possible values for output file library are:

CURLIB	The job's current library is used to locate the output file. If no library is specified as the current library for the job, the QGPL library is used.
LIBL	The library list is used to locate the output file.
Library name	The name of the library where the output file is located.

Restore to ASP device. The name of the auxiliary storage pool (ASP) device to which objects are restored. The default is *SAVASPDEV. The possible values are:

SAVASPDEV	The data is restored to the independent ASP from which it was saved.
ASP device name	The name of the independent ASP where the data will be restored.

Restore to ASP number. The number of the auxiliary storage pool (ASP) to which objects are restored. The default is 0. The possible values are:

0	The data is restored to the ASP from which it was saved. (*SAVASP)
1-32	The number of the ASP where the data will be restored.

Restore to library. The name of the library to which objects are restored. The default is *SAVLIB. The possible values are:

SAVLIB	The data is restored to library from which it was saved.
Library name	The name of the library where the data will be restored.

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

date	The date the objects were saved, in the format CYYMMDD:

	C	Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
	YY	Year
	MM	Month
	DD	Day

Save file. The name and library of the save file that contains the saved data. The first 10 characters contain the save file name; the second 10 characters contain the library where the save file is located.

You can use these special values for the library name:

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

*LIBL   The library list.

Save time. The time the objects were saved. If this key is not specified, the version of the objects to be restored is the first version on the volume.

Note:

1. This key is valid only if the save date key is specified.
2. This key is ignored when the sequence number key is specified.

The possible value is:

time   The time the objects were saved in the format HHMMSS:

        HH   Hour

        MM   Minute

        SS   Second

Saved library. A list of libraries that contain the saved objects. If more than one library is specified, *ALL must be the only object name specified (object information key) and the device cannot be *SAVF. For the format of this key, see "Saved Library Key Format" on page 74.

Sequence number. The sequence number to use for the restore operation when tape is used. The default is -1. The possible values are:

-1   The restore operation searches the tape volume for the file to be restored. (*SEARCH)

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

Spooled file data. A description of spooled file data to be restored. The default is new spooled file data; for each output queue that is restored, spooled file data that was saved with the output queue is restored, if it does not already exist on the system. For the format of this key, see "Spooled File Data Key Format" on page 75.

Type of output information. The type of information that is printed or directed to the output database file. The default is 0. The possible values are:

0   The list contains an entry for each object requested to be restored. (*OBJ)

2   The list contains an entry for each object, database file member, and spooled file requested to be restored. (*MBR)

Volume identifier. The volume identifiers of the tape volumesor optical volumes from which the object data is to be restored. The volume identifiers must be entered in the order in which the data was saved. The default is *MOUNTED. For the format of this field, see "Volume Identifier Format" on page 81.

Allow Object Differences Key Format

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Number in array

Offset			
Dec	**Hex**	**Type**	**Field**
Note: This field repeats for each allow object difference value.			
		CHAR(1)	Allow object difference

Field Descriptions

Allow object difference. Whether differences are allowed between the saved object and the restored object. The differences include:

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

  Note: This key has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.

- File level id: The creation date and time of the database file on the system does no match the creation date and time of the file that was saved.

- Member level id: The creation date and time of the database file member on the system does not match the creation date and time of the member that was saved.

- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.

- Primary group: The primary group of an object on the system is different than the primary group of an object from the save operation.

Note: To specify any value other than 0, you need all object (*ALLOBJ) special authority.

The possible values are:

0   No differences are allowed between the saved object and the restored object. If 0 is specified for the allow object difference field, 1 must be specified for the number in array field. (*NONE)

If an object already exists on the system with a different file level id, member level id, owner, or primary group than the saved object, the object is not restored.

If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored.

If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

1	All differences are allowed between the saved object and the restored object. If 1 is specified for the allow object difference field, 1 must be specified for the number in array field. (*ALL)

If an object already exists on the system with a different owner or primary group than the saved object, the object is restored with the existing values.

If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object.

If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority of the object is set to *EXCLUDE.

If a database file already exists on the system with a different file level id than the saved object, the existing file is renamed and the saved version of the file is restored

If a database file already exists on the system with a different member level id than the saved object, the existing member is renamed and the saved version of the member is restored

2	Authorization list differences are allowed. (*AUTL)

If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object.

If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority of the object is set to *EXCLUDE.

3	Ownership differences are allowed. (*OWNER)

If an object already exists on the system with a different owner than the saved object, the object is restored with the existing value.

4	Primary group differences are allowed. (*PGP)

If an object already exists on the system with a different primary group than the saved object, the object is restored with the existing value.

5	File level id and member level id differences are allowed. (*FILELVL)

If a physical file already exists on the system with a different file level id or member level id than the saved object, but it has the same format level id as the saved object, the data is restored to the existing file.

Number in array. The number of allow object difference values. The possible values are 1 through 4.

Device Key Format

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
Note: This field repeats for each device name.			
		CHAR(10)	Device name

Field Descriptions

Device name. The name of the device used for the restore operation. The possible values for each element of the array are:

*SAVF The restore operation is done using the save file specified by the save file key. If specified, it must be the only element in the array.

MEDDFN			The restore operation is done by using the devices and media that are identified in the media definition, which is specified by the media definition key. If specified, it must be the only element in the array.
Media library device name			The name of the media library device used for the restore operation. If specified, it must be the only element in the array.
Optical device name			The name of the optical device used for the restore operation. If specified, it must be the only element in the array.
Tape device name			The name of the tape device used for the restore operation. A maximum of four tape devices may be used. They must be specified in the order in which they should be used.

Number in array. The number of devices to be used during the restore operation. The possible values are 1 through 4.

File Member Key Format

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Number in array
Note: These fields repeat for each file.			
		CHAR(10)	File name
		CHAR(2)	Reserved
		BINARY(4)	Number of members
Note: This field repeats for each member associated with the given file.			
		CHAR(10)	Member

Field Descriptions

File name. The name of the file being restored. The possible values are:

ALL	The list of member names that follow this value applies to all files indicated in the list of objects to restore. If *ALL is specified for the file name, it must be the only file name in the list.
Database file name	The name of the database file from which the listed members are restored.

Member. The name of the member to restore. The possible values are:

ALL	All members are restored from the specified file. If *ALL is specified for member name, it must be the only member name for that file.
NONE	No members are restored from the specified file. Only the file description is restored.
Member name	The name of the member to restore. It may be either a simple name or a generic name.

Number in array. The number of file and member structures used during the restore operation. The possible values are 1 through 50.

Number of members. The number of member names for the given file name. Possible values are 1 through 50.

Reserved. An ignored field.

Force Object Conversion Key Format

Offset			
Dec	Hex	Type	Field
0	0	CHAR(1)	Convert during restore
1	1	CHAR(1)	Objects to convert

Field Descriptions

Convert during restore. Whether objects should be converted on the restore operation. The possible values are:

0   The objects are not converted during the restore operation. (*NO)

1   The objects are converted during the restore operation. (*YES)

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

2   The objects are converted based on the value of the QFRCCVNRST system value. (*SYSVAL)

      Note: If this value is specified and the system value QFRCCVNRST has a value of 1, the restore operation proceeds as if 1 were specified for the force conversion field and 2 were specified for the objects to convert field.

      If QFRCCVNRST has a value of 0, the restore operation proceeds as if 0 were specified for the force conversion field.

Objects to convert. Which objects should be converted on the restore operation. The default is 2. The possible values are:

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

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

Object Information Key Format

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
Note: These fields repeat for each object name.			
		CHAR(10)	Object name
		CHAR(10)	Object type

Field Descriptions

Number in array. The number of objects that are specified for this key. There is no limit for the number in array field. The total amount of information in the user space, however, cannot exceed 16MB.

Object name. The name of the object that is to be restored. The possible values are:

*ALL   All the objects in the specified libraries, depending on the values specified for object type

Object name Either a simple name or a generic name

Object type. The type of the object that is to be restored. The possible values are:

ALL All objects with the specified object name that are valid types for the RSTOBJ command on the current release of the system.
Object type A valid type for the RSTOBJ command on the current release of the system

Omit Library Key Format

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Number in array
Note: This field repeats for each library name.			
		CHAR(10)	Library name

Field Descriptions

Library name. The name of the library containing the objects to omit. The possible values are:

NONE No libraries are excluded from the restore operation.
Library name Either a simple or generic library name

Number in array. The number of libraries to omit from the restore operation. The possible values are 1 through 300.

Omit Object Information Key Format

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Number in array
Note: These fields repeat for each object name.			
		CHAR(10)	Object name
		CHAR(10)	Library name
		CHAR(10)	Object type

Field Descriptions

Library name. The name of the library that is to be omitted. The possible values are:

ALL All the libraries, depending on the values specified for object and object type
Library name Either a simple name or a generic name

Number in array. The number of values that are specified for this key. The possible values are 1 through 300.

Object name. The name of the object that is to be omitted. The possible values are:

*ALL	All the objects in the specified libraries, depending on the values specified for object type	
Object name	Either a simple name or a generic name	

Object type. The type of the object that is to be omitted. The possible values are:

*ALL	All objects with the specified object name that are valid types for the RSTOBJ command on the current release of the system	
Object type	A valid type for the RSTOBJ command on the current release of the system	

Output Member Key Format

Offset			
Dec	**Hex**	**Type**	**Field**
		CHAR(10)	Output member name
		CHAR(1)	Option

Field Descriptions

Option. An indicator of whether to add to or replace the existing member. The possible values are:

0	The existing records in the specified database file member are replaced by the new records. (*REPLACE)
1	The new records are added to the existing information in the database file member. (*ADD)

Output member name. The name of the file member that receives the output. The possible values are:

*FIRST	The first member in the file is used and receives the output.
Member name	If the member does not exist, the system creates it.

Saved Library Key Format

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Number in array
Note: This field repeats for each library name.			
		CHAR(10)	Library name

Field Descriptions

Library name. The name of the library containing the objects. The possible values are:

*ANY	Restores objects from the first version of all saved libraries found on the tape beginning with the sequence number specified for the sequence number key, or restores objects from all saved libraries found on the optical media in the directory specified for the optical file key.
*SPLF	Spooled file data that was saved with the QSRSAVO API with library name *SPLF specified is to be restored. If this value is specified, it must be the only element in the array, the spooled file data key must be specified, and *ALL must be specified for the object name and object type.

Library name   Either a simple or generic library name

Number in array. The number of libraries used during the restore operation. The possible values are 1 through 300.

Spooled File Data Key Format

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Spooled file data
4	4	BINARY(4)	Length of spooled file data header
8	8	BINARY(4)	Offset to spooled file selection list

Field Descriptions

Length of spooled file data header. The length of the spooled file data header information. The possible values are:

8   The header information ends with the length field.
12   The header information ends with the offset to selection list field.

Offset to spooled file selection list. The offset from the start of the user space to the first "Spooled File Selection List Entry Format." The default is 0. If the value of the spooled file data field is 2, the value of this field must be greater than 0. Otherwise, the value must be 0.

Spooled file data. Whether to save spooled file data and attributes. The default is 3. The possible values are:

0   No spooled file data is restored. (*NONE)
2   Selected spooled file data is restored. The offset to selection list field must be specified.
3   For each output queue that is restored, spooled file data that was saved with the output queue is restored, if it does not already exist on the system. (*NEW)

Spooled File Selection List Entry Format

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Length of spooled file selection list entry
4	4	BINARY(4)	Offset to next spooled file selection list entry
8	8	BINARY(4)	Include or omit
12	C	BINARY(4)	Selection criteria format
16	10	BINARY(4)	Offset to selection criteria
20	14	BINARY(4)	Offset to new attributes

Field Descriptions

Include or omit. Whether the spooled files selected by this entry are included or omitted from the save operation. Omit takes precedence over include. The possible values are:

0   Spooled files that match all of the values specified in the selection criteria are omitted from the restore operation.

1   Spooled files that match all of the values specified in the selection criteria are included in the restore operation, unless another entry omits them. At least one entry must have this value.

Length of spooled file selection list entry. The length of the spooled file selection list entry information. The possible values are:

20   The selection list entry ends with the offset to selection criteria field.

24   The selection list entry ends with the offset to new attributes field.

Offset to new attributes. The offset from the start of the user space to the new attributes for the spooled files included by this selection list entry. The value must be 0 if the Include or omit field value is 0. For the format of the new attributes, see "New Attributes Format" on page 79.

Offset to next spooled file selection list entry. The offset from the start of the user space to the next spooled file selection list entry. The value must be 0 for the last entry in the list.

Offset to selection criteria. The offset from the start of the user space to the selection criteria.

Selection criteria format. The format of the spooled file selection criteria. The possible values are:

1   The selection criteria is specified by the "Spooled File ID Format." This format identifies exactly one spooled file.

2   The selection criteria is specified by the "Spooled File Attributes Format" on page 77. This format identifies any number of spooled files.

Spooled File ID Format

This is the format of the spooled file selection criteria when a value of 1 is specified for the selection criteria format field. The criteria specified must uniquely identify a single spooled file.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Length of Spooled file ID
4	4	CHAR(26)	Qualified job name
30	1E	CHAR(10)	Spooled file name
40	28	BINARY(4)	Spooled file number
44	2C	CHAR(8)	Job system name
52	34	CHAR(7)	Creation date
59	3B	CHAR(6)	Creation time

Field Descriptions

Creation date and time. The date and time the spooled file was created. The date and time must be specified in the format CYYMMDDHHMMSS:

C	Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second

Job system name. The name of the system where the job that created the spooled file ran.

Length of spooled file ID. The length of the spooled file ID information. The possible values are:

65	The spooled file ID ends with the creation time field.

Qualified job name. The name of the job that owns the spooled file. The qualified job name has three parts:

Job name	CHAR(10)	A specific job name.
User name	CHAR(10)	A specific user profile name.
Job number	CHAR(6)	A specific job number.

Spooled file name. The name of the spooled file.

Spooled file number. The unique number of the spooled file. The possible values are:

1-999999	The number of the spooled file for the specified qualified job name and spooled file name.

Spooled File Attributes Format

This is the format of the spooled file selection criteria when a value of 2 is specified for the selection criteria format field.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Length of spooled file attributes
4	4	CHAR(20)	Qualified output queue
24	18	CHAR(10)	Spooled file name
34	22	CHAR(10)	Job name
44	2C	CHAR(10)	User name
54	36	CHAR(6)	Job number
60	3C	CHAR(10)	User-specified data
70	46	CHAR(8)	Job system name
78	4E	CHAR(10)	Form type
88	58	CHAR(13)	Starting creation date and time
101	65	CHAR(13)	Ending creation date and time

Field Descriptions

Ending creation date and time. Spooled files with a creation date and time less than or equal to this date and time are selected. The default is *ALL. The following special value is allowed:

*ALL   Ending creation date and time are not used to select spooled files.

The date and time must be specified in the format CYYMMDDHHMMSS:

C	Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second

Form type. Spooled files with this form type are selected. Either a specific value or generic value may be specified. The default is *ALL. The following special values are allowed:

*ALL   Spooled files with any form type are selected.
*STD   Spooled files that specify the standard form type are selected.

Job name. Spooled files owned by this job are selected. Either a specific name or generic name may be specified. The default is *ALL. The following special value is allowed:

*ALL   Spooled files owned by any job are selected.

Job number. Spooled files owned by a job with this job number are selected. If a job number is specified, then a specific job name and a specific user name must also be specified. The default is *ALL. The following special value is allowed:

*ALL   Spooled files owned by a job with any job number are selected.

Job system name. Spooled files owned by a job on this system are selected. Either a specific name or generic name may be specified. The default is *ALL. The following special values are allowed:

*ALL   Spooled files owned on any system are selected.
*CURRENT Spooled files owned by a job on the current system are selected.

Length of spooled file attributes. The length of the spooled file data attributes information. The possible values are:

24	The spooled file attributes end with the qualified output queue field.
34	The spooled file attributes end with the spooled file name field.
44	The spooled file attributes end with the job name field.
54	The spooled file attributes end with the user name field.
60	The spooled file attributes end with the job number field.
70	The spooled file attributes end with the user-specified data field.
78	The spooled file attributes end with the job system name field.
88	The spooled file attributes end with the form type field.
101	The spooled file attributes end with the starting creation date and time field.
114	The spooled file attributes end with the ending creation date and time field.

Qualified output queue. Spooled files on this output queue are selected, if they were saved with the data selected by the saved library and object information keys. The qualified output queue has two parts:

Object name   CHAR(10). A specific or generic output queue name or the following special value:

                *ALL   Spooled files on all output queues that satisfy the library name are selected.

Library name   CHAR(10). A specific or generic library name, or one of the following special values:

                *ALL   All libraries.

                *CURLIB

                      The job's current library. If no library is specified as the current library for the job, the QGPL library is used.

                *LIBL   The libraries in the library list.

Spooled file name. Spooled files with this name are selected. Either a specific name or generic name may be specified. The default is *ALL. The following special value is allowed:

*ALL   Spooled files with any name are selected.

Starting creation date and time. Spooled files with a creation date and time greater than or equal to this date and time are selected. The default is *ALL. The following special value is allowed:

*ALL   Starting creation date and time are not used to select spooled files.

The date and time must be specified in the format CYYMMDDHHMMSS:

C   Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
YY   Year
MM   Month
DD   Day
HH   Hour
MM   Minute
SS   Second

User name. Spooled files owned by this user are selected. Either a specific name or generic name may be specified. The default is *ALL. The following special value is allowed:

*ALL   Spooled files with any user are selected.

User-specified data. Spooled files with this user-specified data value are selected. Either a specific value or generic value may be specified. The default is *ALL. The following special value is allowed:

*ALL   Spooled files with any user-specified data value are selected.

New Attributes Format

This is the format of new attributes to be assigned to the selected spooled files.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of new attributes
4	4	BINARY(4)	Expiration days
8	8	CHAR(20)	Qualified output queue
28	1C	BINARY(4)	Restore existing spooled file

Field Descriptions

Expiration days. The number of days from the start of the operation when the selected spooled files will expire. The default is 0.

Note: The user needs additional authority to use any value other than 0. The default value of 0 will be used for any spooled files which the user is not authorized to change. The user is authorized to change the expiration date of a spooled file if any of the following conditions are met.

- The user owns the spooled file.
- The user has spool control (*SPLCTL) special authority.
- The user has job control (*JOBCTL) special authority, and the output queue to which the spooled file is being restored is specified as OPRCTL(*YES).
- The user owns the output queue to which the spooled file is being restored, and the output queue is specified as AUTCHK(*OWNER).
- The user has read, add, and delete authorities to the output queue to which the spooled file is being restored, and the output queue is specified as AUTCHK(*DTAAUT).

The possible values are:

-1	The expiration date for the selected spooled files will be set to *NONE (no expiration date).
0	The saved expiration date for the selected spooled files will be used. If a saved expiration date has already passed, a value of -1 will be used.
1-366	The expiration date for the selected spooled files will set to the number of days specified past the date that the restore operation begins.

Length of new attributes. The length of the new attributes information. The possible values are:

8	The new attributes end with the expiration days field.
28	The new attributes end with the qualified output queue field.
32	The new attributes end with the restore existing spooled file field.

Qualified output queue. Spooled files are restored to this output queue, if it is found in the ASP specified for the restore to ASP device key or the restore to ASP number key. The default is *SAME. The qualified output queue has two parts:

Object name	CHAR(10). A specific output queue name or the following special value:
	SAME Spooled files are restored to the output queues from which they were saved. The rest of the qualified output queue must be blank.

Library name   CHAR(10). A specific library name, or blanks when the object name is *SAME, or one of the following special values:

*CURLIB
         The job's current library is used to locate the output queue. If no library is specified as the current library for the job, the QGPL library is used.

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

Restore existing spooled file. Whether to restore a spooled file that already exists on the system. The default is 0. The possible values are:

0   A spooled file that already exists on the system is not restored.
1   If a spooled file already exists on the system, a duplicate copy of the spooled file is restored to the specified output queue with a new creation date and time.

Volume Identifier Format

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
Note: These fields repeat for each volume identifier.			
		BINARY(4)	Length of volume identifier
		CHAR(*)	Volume identifier

Field Descriptions

Length of volume identifier. The character length of the identifier of the volume. The possible value is:

n   The size of a single volume identifier. The maximum size of a tapevolume identifier is 6 characters. The maximum size of an optical volume identifier is 32 characters. If a volume identifier larger than the maximum size is entered for this key, it is truncated to the maximum size. If the volume identifier is *MOUNTED, this value must be 8. If the volume identifier is *SAVVOL, this value must be 7.

Number in array. The number of volume identifiers used during the restore operation. The possible values are 1 through 75.

Volume identifier. The identifier of a volume. The possible values are:

*MOUNTED   The volume currently placed in the device is used. If *MOUNTED is specified, it must be the only value specified. This value cannot be specified for an optical media library device. *MOUNTED cannot be specified for a tape media library device unless a category is set with the Set Tape Category (SETTAPCGY) command.

*SAVVOL   The system, by using the save or restore history information, determines which volumes contain the most recently saved version of the objects. If *SAVVOL is specified, it must be the only value specified. The history information contains only the first 6 characters of any volume name. If the name of an optical volume exceeds 6 characters, you should not use this value.

Volume identifier   The identifier of a volume.

Dependencies between Keys

The following two tables list the dependencies between the different keys. If the dependency holds only for a certain value, then that value is also shown (key = n, where n is the value). Otherwise, if the dependency is true for all values of the key, then only the name of the key is given.

The following table lists the conditions where specifying a certain key forces the use of another key.

If you specify...	...must be specified
More than one library name, Generic library name, or Library name = *ANY	Object name = *ALL Device <> *SAVF Restore library = *SAVLIB[1] Label = *SAVLIB[1] Optical file = '*'[1] or 'directory/*'
Device = tape device	Volume identifier [1] Sequence number [1] Label [1] End of media option [1]
Device = optical device	Volume identifier Optical file [1]
Device = media definition	Media definition
Output = 1	Type of output information = 0[1]
Output = 2	Output file Output member [1] Type of output information [1]
Save time	Save date
Volume identifier = *SAVVOL	Label = *SAVLIB[1]
Library name = *SPLF	Object name = *ALL[1] Object type = *ALL[1] Spooled file data = 2
Spooled file data = 2	Object name = *ALL[1] Object type = *ALL[1] Library name = *SPLF or Object type = *OUTQ Library name <> *SPLF
Notes: 1. This key does not have to be explicitly specified. The default may be taken to satisfy this dependency.	

The following table lists the conditions where specifying a certain key excludes the user from using another key, or a particular value of that key.

If you specify...	...cannot be specified
Save file	Volume identifier Sequence number Label End of media option Optical file Media definition
Media definition	Volume identifier Sequence number Optical file

If you specify...	...cannot be specified
Tape, optical, or media definition for the device	Save file
Output = 0	Output file Output member Type of output information
Optical file	Label Sequence number
Allow object differences = 1	Database member option = 0
Volume identifier = *SAVVOL	Sequence number
Database member option = 0	File member
Restore ASP	Restore ASP device name

Relationship to RSTOBJ Command

Because of the relationship between the QSRRSTO API and the RSTOBJ command, the following situations should be noted:

- Message text: Several messages produced by this API refer to parameters or values of the RSTOBJ command. To determine which key a given parameter corresponds to, see "Valid Keys" on page 64. To determine which key value a given parameter value corresponds to, see "Field Descriptions" on page 65.
- Command type: The command type listed for the API on headings of displays and print files is RSTOBJ, not QSRRSTO.

Error Messages

Message ID	Error Message Text
CPF222E E	&1 special authority is required.
CPF24B4 E	Severe error while addressing parameter list.
CPF3700 E	All CPF37xx messages could be signalled. xx is from 01 to FF.
CPF3800 E	All CPF38xx messages could be signalled. xx is from 01 to FF.
CPF3C31 E	Object type &1 is not valid.
CPF3C4D E	Length &1 for key &2 not valid.
CPF3C81 E	Value for key &1 not valid.
CPF3C82 E	Key &1 not valid for API &2.
CPF3C83 E	Key &1 not allowed with value specified for key &2.
CPF3C84 E	Key &1 required with value specified for key &2.
CPF3C85 E	Value for key &1 not allowed with value for key &2.
CPF3C86 E	Required key &1 not specified.
CPF3C87 E	Key &1 allows one value with special value.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF5729 E	Not able to allocate object &1.
CPF9800 E	All CPF98xx messages could be signaled. xx is from 01 to FF.
CPFB8ED E	Device description &1 not correct for operation.

≪ API introduced: V5R4

Retrieve Backup Detail (QEZRTBKD) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Object name	Input	Char(*)
4	Object name length	Input	Binary(4)
5	Format name	Input	Char(8)
6	Object type	Input	Char(10)
7	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Retrieve Backup Detail (QEZRTBKD) API retrieves more detailed information about the library or folder that is to be backed up.

Authorities and Locks

Backup Object
        *USE

Required Parameter Group

Receiver variable
        OUTPUT; CHAR(*)

        The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
        INPUT; BINARY(4)

        The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Object name
        INPUT; CHAR(*)

        The name of the object to retrieve backup detail information about. The length of this field is based on the Object type parameter and the Object name length parameter. Specify either a 10-character library name or specify a 12-character folder name.

Object name length
        INPUT; BINARY(4)

        The length of the name of the object about which to retrieve backup detail information. Allowable object name lengths follow:

10   The 10-character library name when the object type parameter is *LIB.
12   The 12-character folder name when the object type parameter is *FLR.

Format name
        INPUT; CHAR(8)

        The name of the format to be used to return information to caller. The following format may be used:

RBKD0100   Backup detail information. For more information, see "RBKD0100 Format."

Object type
    INPUT; CHAR(10)

    The type of object for which you are requesting information. Allowable object types follow:

*FLR   Folder
*LIB   Library

Error code
    I/O; CHAR(*)

    The structure in which to return error information. For the format of the structure, see Error code parameter.

RBKD0100 Format

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Bytes available
4	4	BINARY(4)	Bytes returned
8	8	CHAR(7)	Last saved date
15	F	CHAR(6)	Last saved time
21	15	CHAR(50)	Object description text
71	47	CHAR(1)	Changed since last backup

Field Descriptions

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Changed since last backup. Whether the object changed since the last backup. Possible values follow:

0   No change has been made since the last backup.
1   Change has been made since the last backup.

Last saved date. The date that the object was last saved to media. The format of this field is in the CYYMMDD as follows:

C   Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
YY   Year
MM   Month
DD   Day

Last saved time. The time that the object was last saved to media. The format of this field is in the HHMMSS as follows:

HH   Hour

MM   Minute
SS   Second

Object description text. The text that describes the object.

Error Messages

Message ID	Error Message Text
CPF1EC7 E	The length &1 is not valid when the specified type is &2.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C31 E	Object type &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF8148 E	Object &4 type &3 in &9 damaged.
CPF8A75 E	Not authorized to access folder &1.
CPF8A77 E	Folder &1 not found.
CPF8A78 E	Folder &1 in use.
CPF8A79 E	Folder &1 is logically damaged.
CPF8A80 E	Document &2 in use in folder &1.
CPF9012 E	Start of document interchange session not successful for &1.
CPF9032 E	Document interchange session not started.
CPF9076 E	Internal system error processing documents or folders.
CPF9095 E	Folder &1 is damaged.
CPF909A E	Document &2 in folder &1 is damaged.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9830 E	Cannot assign library &1.
CPF9838 E	User profile storage limit exceeded.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V3R7

Retrieve Backup History (QEZRTBKH) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Retrieve Backup History (QEZRTBKH) API retrieves information about the backup status and history into a single variable in the calling program. The amount of information that is returned depends on the size of the variable. The information that this API returns is the same information that the Display Backup Status (DSPBCKSTS) command returns.

Authorities and Locks

User Index Authority
    *USE

User Index Lock
    *SHRRD

Required Parameter Group

Receiver variable
    OUTPUT; CHAR(*)

    The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
    INPUT; BINARY(4)

    The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
    INPUT; CHAR(8)

    The format of the command information to be returned. One of the following format names may be used:

RBKH0100	Basic backup status and history. For more information, see "RBKH0100 Format."
RBKH0200	Detailed backup status and information. For more information, see "RBKH0200 Format" on page 88.

Error code
    I/O; CHAR(*)

    The structure in which to return error information. For the format of the structure, see Error Code Parameter.

RBKH0100 Format

The following table describes the information that is returned in the receiver variable for the RBKH0100 format. For detailed descriptions of the fields, see "Field Descriptions" on page 89.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(7)	Last date all user libraries backed up
15	F	CHAR(6)	Last backup time of all user libraries
21	15	CHAR(4)	Tape set for all user libraries
25	19	CHAR(7)	Last date all user libraries backed up (changes only)
32	20	CHAR(6)	Last backup time of all user libraries (changes only)
38	26	CHAR(4)	Tape set for all user libraries (changes only)
42	2A	CHAR(7)	Last date libraries on list backed up

Offset		Type	Field
Dec	Hex		
49	31	CHAR(6)	Last backup time of libraries on list
55	37	CHAR(4)	Tape set for libraries on list
59	3B	CHAR(7)	Last date libraries on list backed up (changes only)
66	42	CHAR(6)	Last backup time of libraries on list (changes only)
72	48	CHAR(4)	Tape set for libraries on list (changes only)
76	4C	CHAR(7)	Last date all folders backed up
83	53	CHAR(6)	Last backup time of all folders
89	59	CHAR(4)	Tape set for folders
93	5D	CHAR(7)	Last date all folders backed up (changes only)
100	64	CHAR(6)	Last backup time of all folders (changes only)
106	6A	CHAR(4)	Tape set for folders (changes only)
110	6E	CHAR(7)	Last date folders on list backed up
117	75	CHAR(6)	Last backup time of folders on list
123	7B	CHAR(4)	Tape set for folders on list
127	7F	CHAR(7)	Last date security data backed up
134	86	CHAR(6)	Last backup time of security data
140	8C	CHAR(4)	Tape set for security data
144	90	CHAR(7)	Last date configuration data backed up
151	97	CHAR(6)	Last backup time of configuration
157	9D	CHAR(4)	Tape set for configuration data
161	A1	CHAR(7)	Last date calendars backed up
168	A8	CHAR(6)	Last backup time of calendars
174	AE	CHAR(4)	Tape set for calendars
178	B2	CHAR(7)	Last date mail backed up
185	B9	CHAR(6)	Last backup time of mail
191	BF	CHAR(4)	Tape set for mail
195	C3	CHAR(7)	Last date all user directories backed up
202	CA	CHAR(6)	Last backup time of all user directories
208	D0	CHAR(4)	Tape set for all user directories backed up
212	D4	CHAR(7)	Last date all user directories backed up (changes only)
219	DB	CHAR(6)	Last backup time of all user directories (changes only)
225	E1	CHAR(4)	Tape set for all user directories backed up (changes only)
229	E5	CHAR(21)	Reserved

RBKH0200 Format

The following table describes the information that is returned in the receiver variable for the RBKH0200 format. For detailed descriptions of the fields, see "Field Descriptions" on page 89.

Offset		Type	Field
Dec	Hex		
0	0		Returns everything from format RBKH0100
250	FA	BINARY(4)	Number of backup date entries
254	FE	BINARY(4)	Length of a backup date entry
Note: Format of a backup date entry. The backup date entry fields repeat, in the order listed, by the number of backup date entries. The decimal and hexadecimal offsets to the following fields depend on the number of backup date entries and the length of a backup date entry.			
		CHAR(7)	Backup date
		CHAR(6)	Backup time
		CHAR(10)	Backup options
		CHAR(4)	Tape set
		CHAR(1)	Changes only
		CHAR(1)	User libraries saved
		CHAR(1)	Folders saved
		CHAR(1)	User directories saved
		CHAR(1)	Security data saved
		CHAR(1)	Configuration saved
		CHAR(1)	Calendars saved
		CHAR(1)	Mail saved

Field Descriptions

Some of the fields use a date format (CYYMMDD) as follows:

C	Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
YY	Year
MM	Month
DD	Day

Some of the fields use a time format (HHMMSS), where:

HH	Hour
MM	Minute
SS	Second

For more information on the following fields, refer to the documentation for the Display Backup Status (DSPBCKSTS) command in the Control Language (CL) information.

Backup date. The completion date of the backup. This date is in the format CYYMMDD.

Backup time. The completion time of the backup. This time is in the format HHMMSS.

Backup options. The backup option that were used. The possible values are:

*DAILY
*WEEKLY
*MONTHLY

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Calendars saved. Whether OfficeVision calendars were saved during the backup.

0 No OfficeVision calendars were saved during the backup.
1 All OfficeVision calendars were saved during the backup.

Changes only. Whether the backup saved only the changes to libraries, directories, and folders.

0 All objects in the libraries, folders, and directories were saved during the backup.
1 Only changes made to the libraries, folders, and directories since the last backup were saved during the backup.

Configuration saved. Whether the system configuration was saved.

0 The system configuration was not saved during the backup.
1 The system configuration was saved during the backup.

Folders saved. Which folders were saved during the backup. The possible values follow:

1 Only folders that were selected from the folder backup list were saved.
2 All folders were saved.
3 No folders were saved.

Last backup time of calendars. The completion time of the most recent backup of all the calendars for OfficeVision. This time uses the 24-hour clock and is in the format HHMMSS.

Last backup time of mail. The completion time of the most recent backup of all the mail for OfficeVision. This time uses the 24-hour clock and is in the format HHMMSS.

Last backup time of all folders. The completion time of the most recent backup of all the objects in all the root folders. This time uses the 24-hour clock and is in the format HHMMSS.

Last backup time of all folders (changes only). The completion time of the most recent backup of all the objects in all the root folders that changed since the previous backup. This time uses the 24-hour clock and is in the format HHMMSS.

Last backup time of all user directories. The completion time of the most recent backup of all the objects in all the user directories on the system. This time uses the 24-hour clock and is in the format HHMMSS.

Last backup time of all user directories (changes only). The completion time of the most recent backup of all the objects in all the user directories that changed since the previous backup. This time uses the 24-hour clock and is in the format HHMMSS.

Last backup time of all user libraries. The completion time of the most recent backup of all the objects in all the user libraries on the system. This time uses the 24-hour clock and is in the format HHMMSS.

Last backup time of all user libraries (changes only). The completion time of the most recent backup of all the objects in all the user libraries that changed since the previous backup. This time uses the 24-hour clock and is in the format HHMMSS.

Last backup time of configuration. The completion time of the most recent backup of the system configuration. This uses the 24-hour clock and is in the format HHMMSS.

Last backup time of folders on list. The completion time of the most recent backup of all the objects in all the folders on the folder backup list. This time uses the 24-hour clock and is in the format HHMMSS.

Last backup time of libraries on list. The completion time of the most recent backup of all the objects in all the libraries on the library backup list. This time uses the 24-hour clock and is in the format HHMMSS.

Last backup time of libraries on list (changes only). The completion time of the most recent backup of all the objects in all the libraries on the library backup list that changed since the previous backup. This time uses the 24-hour clock and is in the format HHMMSS.

Last backup time of security data. The completion time of the most recent backup of all the system security data. This uses the 24-hour clock and is in the format HHMMSS.

Last date all folders backed up. The completion date for the most recent backup of all the objects in all the root folders on the system. This date is in the format CYYMMDD.

Last date all folders backed up (changes only). The completion date for the most recent backup of all the objects in all the folders that changed since the previous backup. This date is in the format CYYMMDD.

Last date all user directories backed up (changes only). The completion date for the most recent backup of all the objects in all the user directories that changed since the previous backup. This date is in the format CYYMMDD.

Last date all user directories backed up. The completion date for the most recent backup of all the objects in all the user directories on the system. This date is in the format CYYMMDD.

Last date all user libraries backed up. The completion date for the most recent backup of all the objects in all the user libraries on the system. This date is in the format CYYMMDD.

Last date all user libraries backed up (changes only). The completion date for the most recent backup of all the objects in all the user libraries that changed since the previous backup. This date is in the format CYYMMDD.

Last date calendars backed up. The completion date for the most recent backup of all the OfficeVision calendars on the system. This date is in the format CYYMMDD.

Last date configuration data backed up. The last backup date for the most recent backup of the system configuration.

Last date folders on list backed up. The completion date for the most recent backup of all the objects in all the folders in the folder backup list. This date is in the format CYYMMDD.

Last date libraries on list backed up. The completion date for the most recent backup of all the objects in all the libraries in the library backup list. This date is in the format CYYMMDD.

Last date libraries on list backed up (changes only). The completion date for the most recent backup of all the objects in all the libraries in the library backup list that changed since the previous backup. This date is in the format CYYMMDD.

Last date mail backed up. The completion date for the most recent backup of all the OfficeVision mail on the system. This date is in the format CYYMMDD.

Last date security data backed up. The last backup date for the most recent backup of system security data.

Length of a backup date entry. The length of one backup date entry.

Mail saved. Whether mail was saved during the backup.

| 0 | OfficeVision mail was not saved during the backup. |
| 1 | OfficeVision mail was saved during the backup. |

Number of backup date entries. The number of backup date e ntries that are contained in format RBKH0200.

Reserved. This space is reserved for future use.

Security data saved. Whether the security data was saved.

| 0 | Security data was not saved during the backup. |
| 1 | Security data was saved during the backup. |

Tape set. The name of the last tape set that the system used to complete the backup of the information described in this backup date entry.

Tape set for all user directories backed up. The name of the last tape set that the system used for the most recent backup of all the objects in all the user directories on the system. The possible value follows:

| *ANY | The option that the system used for the backup did not specify a tape set name. |

Tape set for all user directories backed up (changes only). The name of the last tape set that the system used for the most recent backup of all the objects in all the user directories that changed since the previous backup. The possible value follows:

| *ANY | The option that the system used for the backup did not specify a tape set name. |

Tape set for all user libraries. The name of the last tape set that the system used for the most recent backup of all the objects in all the user libraries on the system. The possible value follows:

| *ANY | The option that the system used for the backup did not specify a tape set name. |

Tape set for all user libraries (changes only). The name of the last tape set that the system used for the most recent backup of all the objects in all the user libraries that changed since the previous backup. The possible value follows:

| *ANY | The option that the system used for the backup did not specify a tape set name. |

Tape set for calendars. The name of the last tape set that the system used for the most recent backup of all the calendars for OfficeVision.

The possible value follows:

*ANY   The option that the system used for the backup did not specify a tape set name.

Tape set for configuration. The name of the last tape set that the system used for the most recent backup of the system configuration. The possible value follows:

*ANY   The option that the system used for the backup did not specify a tape set name.

Tape set for folders. The name of the last tape set that the system used for the most recent backup of all the objects in all the root folders on the system. The possible value follows:

*ANY   The option that the system used for the backup did not specify a tape set name.

Tape set for folders (changes only). The name of the last tape set that the system used for the most recent backup of all the new and changed documents from all the new and changed root folders since the previous backup. The possible value follows:

*ANY   The option that the system used for the backup did not specify a tape set name.

Tape set for folders on list. The name of the last tape se t that the system used for the most recent backup of all the folders that were selected for backup from the folder backup list. The possible value follows:

*ANY   The option that the system used for the backup did not specify a tape set name.

Tape set for libraries on list. The name of the last tape set that the system used for the most recent backup of all the libraries on the library backup list. The possible value follows:

*ANY   The option that the system used for the backup did not specify a tape set name.

Tape set for libraries on list (changes only). The name of the last tape set that the system used for the most recent backup of all the objects in the libraries on the library backup list that changed since the previous backup. The possible value follows:

*ANY   The option that the system used for the backup did not specify a tape set name.

Tape set for mail. The name of the last tape set that the system used for the most recent backup of all the mail for OfficeVision. The possible value follows:

*ANY   The option that the system used for the backup did not specify a tape set name.

Tape set for security data. The name of the last tape set that the system used for the most recent backup of the system security data. The possible value follows:

*ANY   The option that the system used for the backup did not specify a tape set name.

User directories saved. The directories that were saved during the backup. The possible values follow:

| 2 | All user directories were saved. |
| 3 | No user directories were saved. |

User libraries saved. The libraries that were saved during the backup. The possible values follow:

1	Only user libraries that were selected from the library backup list were saved.
2	All user libraries were saved.
3	No user libraries were saved.

Error Messages

Message ID	Error Message Text
CPF1E00 E	All CPF1Exx messages could be returned. xx is from 01 to FF.
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

Retrieve Backup Options (QEZRTBKO) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Backup option	Input	Char(10)
5	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Retrieve Backup Options (QEZRTBKO) API returns in a receiver variable the backup options for the requested backup type. If you request all options, the receiver variable contains the header, which is followed by three RBKO0100 formats, one for each backup type.

Authorities and Locks

User Index Authority
       *USE

User Index Lock
       *SHRRD

Required Parameter Group

Receiver variable
       OUTPUT; CHAR(*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
INPUT; CHAR(8)

The format of the backup option descriptions to be returned. The valid format names are RBKO0100 and RBKO0200.

RBKO0100	This format returns information about what the user has selected to be saved on the next backup for that type (*DAILY, *WEEKLY, or *MONTHLY).
RBKO0200	This format returns information on the last backup date and time, and when the next backup date and time for that backup option are scheduled to occur.

Backup option
INPUT; CHAR(10)

The backup options to retrieve. Possible values follow:

DAILY	Daily backup options.
WEEKLY	Weekly backup options.
MONTHLY	Monthly backup options.
ALL	The backup options for all types of backup (*DAILY, *WEEKLY, and *MONTHLY).

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

RBOH0100 Format

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to daily options
12	C	BINARY(4)	Offset to weekly options
16	10	BINARY(4)	Offset to monthly options

RBKO0100 Format

Note: The following is accessed by using the offsets that are provided in format RBOH0100.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Offset to list of backup device names
4	4	BINARY(4)	Number of backup devices
8	8	BINARY(4)	Offset to list of tape sets to rotate
12	C	BINARY(4)	Number of tape sets to rotate
16	10	CHAR(4)	Last tape set used
20	14	CHAR(4)	Next tape set to be used
24	18	CHAR(1)	Erase tape before backup
25	19	CHAR(1)	Back up user libraries
26	1A	CHAR(1)	Back up folders
27	1B	CHAR(1)	Back up user directories
28	1C	CHAR(1)	Back up security data
29	1D	CHAR(1)	Back up configuration data
30	1E	CHAR(1)	Back up OfficeVision mail
31	1F	CHAR(1)	Back up OfficeVision calendars
32	20	CHAR(1)	Submit as batch job
33	21	CHAR(1)	Save changed objects only
34	22	CHAR(1)	Print detailed report
35	23	CHAR(10)	User exit program name
45	2D	CHAR(10)	User exit program library name
55	37	CHAR(1)	Reserved
56	38	BINARY(4)	Offset to additional information
The offsets to these fields depend on the number of backup devices and the number of tape sets to rotate.		ARRAY OF CHAR(10)	Backup devices
		ARRAY OF CHAR(4)	Tape sets to rotate

RBKO0200 Format

Offset			
Dec	Hex	Type	Field
0	0	CHAR(*)	All data in format RBKO0100
The offsets to these fields depend on the number of backup devices and the number of tape sets to rotate.		CHAR(7)	Last backup date
		CHAR(6)	Last backup time
		CHAR(7)	Next backup date
		CHAR(6)	Next backup time

Field Descriptions

Back up configuration data. Whether to save the system configuration data. Possible values follow:

0	Do not save configuration data.
1	Save configuration data.

Backup devices. The tape devices that are used for a backup.

Back up folders. Which root folders are backed up. Possible values follow:

1	The root folders that are selected for backup in the folder backup list are backed up.
2	All root folders are backed up.
3	No root folders are backed up.

Back up OfficeVision calendars. Whether to save OfficeVision calendar data. OfficeVision calendars are also saved when QUSRSYS is saved. Possible values follow:

0	Calendars are not saved when the backup is run.
1	Calendars are saved when the backup is run.
9	Calendars are not saved when the backup is run because the OfficeVision calendar function is not available.

Back up OfficeVision mail. Whether to save OfficeVision mail. The backup ignores this option if FLR(*ALL) is specified.

0	Mail is not saved when the backup is run.
1	Mail is saved when the backup is run.
9	Mail is not saved when the backup is run because the OfficeVision mail function is not available.

Back up security data. Whether to save the system security data. Possible values follow:

0	Security data is not saved when the backup is run.
1	Security data is saved when the backup is run.

Back up user libraries. Which user libraries are saved when a backup is run. For information on user libraries, see *ALLUSR in Generic library names.

Possible values follow:

1	The objects in the libraries that are selected for backup from the library backup list are backed up.
2	All user libraries are saved when the backup is run.
3	No user libraries are saved when the backup is run.

Back up user directories. Whether to save all user directories. This does not include any IBM-supplied directories that contain user data. Some directories that are not saved include QOPT, QFileSvr.400, QSYS.LIB, and QDLS (QDLS data is saved under the option to save folders). Possible values follow:

2	All user directories are saved when the backup is run.
3	Do not back up any user directories.

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Erase tape before backup. Whether to clear the tape and start the save operation at sequence number 1. Possible values follow:

0   The save operation continues at the next available sequence number.
1   The tape is cleared, and the save operation starts at sequence number 1.

Last backup date. The completion date of the last Operational Assistant backup in CYYMMDD format.

Last backup time. The completion time of the last Operational Assistant backup in HHMMSS format.

Last tape set used. The tape set that was used to complete the last Operational Assistant backup.

Next backup date. The next date that an Operational Assistant backup is run in CYYMMDD format.

Next backup time. The time of the next scheduled backup. This time is in the format HHMMSS and uses the 24-hour clock.

Next tape set to be used. The tape set that is used to start the next backup.

Number of backup devices. The number of backup devices that are defined as available for use in an Operational Assistant backup.

Number of tape sets to rotate. The number of tape sets that are defined to be used in a backup.

Offset to additional information. The offset from the start of format RBOH0100 to the last backup date field.

Offset to daily options. The offset from the start of the RBOH0100 structure to the backup options.

Offset to list of backup device names. The offset from the start of the RBOH0100 structure to the list of device names. This list is defined by the user to be used during an Operational Assistant backup.

Offset to list of tape sets to rotate. The offset from the start of the RBOH0100 structure to the list of tape sets to be used in an Operational Assistant backup.

Offset to monthly options. The offset from the start of the RBOH0100 structure to the monthly backup options.

Offset to weekly options. The offset from the start of the RBOH0100 structure to the weekly backup options.

Print detailed reports. Whether each save command that supports OUTPUT(*PRINT) generates a detailed list of the objects that were saved in addition to a summary report. Possible values follow:

0   Only a summary report is generated from the backup. A detailed report is not generated.
1   A detailed list of saved objects is printed. A summary report is always printed.

Reserved. This field is ignored.

Save changed objects only. Whether to save only changed objects in the libraries, folders, and directories being backed up. Possible values follow:

0   All of the objects in the requested libraries, folders, and directories are backed up.
1   Only objects that were changed since the last backup are saved.

Submit as batch job. Whether the backup is scheduled to be submitted as a batch job. Possible values follow:

0   The backup is run interactively.
1   The backup is submitted as a batch job.

Tape sets to rotate. The name of the one or more tape sets to be used for backup rotation.

User exit program library name. The name of the library that contains the user exit program.

blank   No exit program is defined so no library name is required.
*LIBL   The library list is searched to find the exit program.

User exit program name. This exit program is run before and after a backup is run to allow users to customize their backups.

*NONE   No exit program is run.

Error Messages

Message ID	Error Message Text
CPF1EC5 E	Backup option &1 is not valid.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

Retrieve Backup Schedule (QEZRTBKS) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Retrieve Backup Schedule (QEZRTBKS) API returns in a receiver variable information about when the Operational Assistant backups are scheduled to be run.

Authorities and Locks

User Index Authority
      *USE

Job Schedule Entry Authority
        *USE

User Index Lock
        *SHRRD

Job Schedule Lock
        *SHRRD

Required Parameter Group

Receiver variable
        OUTPUT; CHAR(*)

  The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
        Input; BINARY(4)

  The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
        INPUT; CHAR(8)

  The name of the format in which to return the backup schedule. The following format name may be used:

RBKS0100   Basic schedule information. For more information, see "RBKS0100 Format."

Error code
        I/O; CHAR(*)

  The structure in which to return error information. For the format of the structure, see Error Code Parameter.

RBKS0100 Format

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Hours before backup to send tape message
12	C	BINARY(4)	Occurrence in month to run backup
16	10	CHAR(1)	Run backup using this schedule
17	11	CHAR(1)	Sunday backup
18	12	CHAR(6)	Sunday backup time
24	18	CHAR(1)	Monday backup
25	19	CHAR(6)	Monday backup time
31	1F	CHAR(1)	Tuesday backup
32	20	CHAR(6)	Tuesday backup time

Offset		Type	Field
Dec	Hex		
38	26	CHAR(1)	Wednesday backup
39	27	CHAR(6)	Wednesday backup time
45	2D	CHAR(1)	Thursday backup
46	2E	CHAR(6)	Thursday backup time
52	34	CHAR(1)	Friday backup
53	35	CHAR(6)	Friday backup time
59	3B	CHAR(1)	Saturday backup
60	3C	CHAR(6)	Saturday backup time

Field Descriptions

Some of the fields use a time format (HHMMSS), where:

HH	Hour
MM	Minute
SS	Second

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Friday backup. The type of backup that is run on Friday. Possible values follow:

blank	No backup is scheduled for this day of the week.
1	A daily backup is run on this day.
2	A weekly backup is run on this day.
3	A monthly backup is run on this day.
4	A *WEEKMONTH backup is run on this day. A weekly backup is run on this day of the week every week except on the week that a monthly backup is run. In that case, the monthly backup is run.

Friday backup time. Using the 24-hour clock, the time that the backup takes place (in format HHMMSS).

Hours before backup to send tape message. If you choose, you may have the system send a reminder to the system operator to load the tape before a backup starts. Specify the number of hours before the backup that you would like this message to be sent. Possible values follow:

0	No message is sent to the system operator before the backup.
1-24	The number of hours prior to backup to send the message.

Monday backup. The type of backup that is run on Monday. Possible values follow:

blank	No backup is scheduled for this day of the week.
1	A daily backup is run on this day.
2	A weekly backup is run on this day.
3	A monthly backup is run on this day.
4	A *WEEKMONTH backup is run on this day. A weekly backup is run on this day of the week every week except on the week that a monthly backup is run. In that case, the monthly backup is run.

Monday backup time. Using the 24-hour clock, the time that the backup takes place (in format HHMMSS).

Occurrence in month to run backup. The week of the month that you want the backup to occur when the backup type is either *MONTHLY or *WEEKMONTH. Possible values follow:

0	No monthly backup is scheduled to run.
1-4	A value 1 through 4 corresponds to the week of the month that the monthly backup is run.
5	The monthly backup is run on the last week of the month.

Run backup using this schedule. Whether you want to use the backup schedule. Possible values follow:

0	A schedule has been created, but it is not being used at this time.
1	Backups are run according to the schedule.

Saturday backup. The type of backup that is to run on Saturday. Possible values follow:

blank	No backup is scheduled for this day of the week.
1	A daily backup is run on this day.
2	A weekly backup is run on this day.
3	A monthly backup is run on this day.
4	A *WEEKMONTH backup is run on this day. A weekly backup is run on this day of the week every week except on the week that a monthly backup is run. In that case, the monthly backup is run.

Saturday backup time. Using the 24-hour clock, the time that the backup takes place (in HHMMSS).

Sunday backup. The type of backup that is to be run on Sunday. Possible values follow:

blank	No backup is scheduled for this day of the week.
1	A daily backup is run on this day.
2	A weekly backup is run on this day.
3	A monthly backup is run on this day.
4	A *WEEKMONTH backup is run on this day. A weekly backup is run on this day of the week every week except on the week that a monthly backup is run. In that case, the monthly backup is run.

Sunday backup time. Using the 24-hour clock, the time that the backup takes place (in format HHMMSS).

Thursday backup. The type of backup that is run on Thursday. Possible values follow:

blank	No backup is scheduled for this day of the week.
1	A daily backup is run on this day.
2	A weekly backup is run on this day.
3	A monthly backup is run on this day.
4	A *WEEKMONTH backup is run on this day. A weekly backup is run on this day of the week every week except on the week that a monthly backup is run. In that case, the monthly backup is run.

Thursday backup time. Using the 24-hour clock, the time that the backup takes place (in format HHMMSS).

Tuesday backup. The type of backup that is run on Tuesday. Possible values follow:

blank	No backup is scheduled for this day of the week.
1	A daily backup is run on this day.
2	A weekly backup is run on this day.
3	A monthly backup is run on this day.
4	A *WEEKMONTH backup is run on this day. A weekly backup is run on this day of the week every week except on the week that a monthly backup is run. In that case, the monthly backup is run.

Tuesday backup time. Using the 24-hour clock, the time that the backup takes place (in format HHMMSS).

Wednesday backup. The type of backup that is run on Wednesday. Possible values follow:

blank	No backup is scheduled for this day of the week.
1	A daily backup is run on this day.
2	A weekly backup is run on this day.
3	A monthly backup is run on this day.
4	A *WEEKMONTH backup is run on this day. A weekly backup is run on this day of the week every week except on the week that a monthly backup is run. In that case, the monthly backup is run.

Wednesday backup time. Using the 24-hour clock, the time that the backup takes place (in format HHMMSS).

Error Messages

Message ID	Error Message Text
CPF1629 E	Not authorized to job schedule &1.
CPF1632 E	Job schedule entry &3 number &4 damaged.
CPF1637 E	Job schedule &1 in library &2 in use.
CPF1640 E	Job schedule &1 in library &2 does not exist.
CPF1641 E	Job schedule &1 in library &2 damaged.
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF812C E	Job schedule &4 in &9 damaged.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

Retrieve Cartridge Filter (QTARCTGF) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

The Retrieve Cartridge Filter (QTARCTGF) API retrieves the currently defined cartridge filter for the system.

Authorities and Locks

None.

Required Parameter Group

Receiver variable
> OUTPUT; CHAR(*)

> The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
> INPUT; BINARY(4)

> The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
> INPUT; CHAR(8)

> Specifies the content and format of the information being returned.

> The RFTR0100 format is used for the cartridge filter information. See "RFTR0100 Format" to view the information returned for this format.

Error code
> I/O; CHAR(*)

> The structure in which to return error information. For the format of the structure, see Error code parameter.

RFTR0100 Format

The following table shows the information that is returned for the RFTR0100 format. For more details about the fields in the following table, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to array
12	C	BINARY(4)	Number of array entries
16	10	BINARY(4)	Length of array entry
		Array(*) of CHAR(*)	Cartridge filter array

Field Descriptions

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Cartridge filter array. An array of cartridge filters currently defined for the system. Each array entry may consist of either a specific cartridge identifier, or a generic name of a cartridge identifier. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. A generic name specifies all cartridge identifiers with names that begin with the generic prefix.

Note: The complete cartridge filter array will not be returned if the receiver variable is not large enough.

Length of array entry. The length, in bytes, of a single cartridge filter entry.

Number of array entries. The number of cartridge filter entries that were returned. A value of zero is returned if there is no cartridge filter set.

Offset to array. The offset, in bytes, to the cartridge filter array. A value of zero is returned if there is no cartridge filter set.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of receiver variable is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF6708 E	Command ended due to error.
CPF67C9 E	An Error occured during a cartridge filter operation.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R3

Retrieve Cartridge Information (QTARCTGI) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Device description	Input	Char(10)
5	Requested cartridge ID	Input	Char(6)
6	Requested category	Input	Char(18)
7	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Retrieve Cartridge Information (QTARCTGI) API retrieves a list of the cartridges in a tape library device and their attributes. This API provides a function similar to the Display Tape Cartridge (DSPTAPCTG) command.

Authorities and Locks

Device Description Authority
　　　*USE

Required Parameter Group

Receiver variable
　　　OUTPUT; CHAR(*)

　　　The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

　　　Note: With the exception of the first cartridge information entry, only complete cartridge information entries will be returned.

Length of receiver variable
　　　INPUT; BINARY(4)

　　　The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
　　　INPUT; CHAR(8)

　　　The content and format of the information being returned.

　　　The RCTG0100 format must be used for the cartridge information. See "RCTG0100 Format" on page 107 to view the information returned for this format.

Device description
　　　INPUT; CHAR(10)

　　　The name of the tape library device containing the cartridge(s) for which the data is to be returned.

Requested cartridge ID
　　　INPUT; CHAR(6)

　　　The cartridge or cartridges for which the data is to be returned. This parameter is used in conjunction with the Requested category parameter to filter the data that is returned. This parameter can contain a specific cartridge identifier, or a generic name of a cartridge identifier. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. A generic name specifies all cartridge identifiers with names that begin with the generic prefix. The following special value is also supported:

　　　*ALL　　Information will be returned for all cartridges in the specified tape library device that are in the requested category.

Requested category
　　　INPUT; CHAR(18)

　　　The category name and system for which the data is to be returned. This parameter is used in conjunction with the Requested cartridge parameter to filter the data that is returned. The requested category has two parts:

　　　Category name
　　　　　　CHAR(10). A specific category name or one of the following special values:

ALL All categories that are owned by the specified category system are searched for the requested cartridge indentifiers.

NOSHARE
> The *NOSHARE category that is owned by the specified category system is searched for the requested cartridge identifiers.

IPL The *IPL category that is owned by the specified category system is searched for the requested cartridge identifiers.

NL The *NL category that is owned by the specified category system is searched for the requested cartridge identifiers.

SYSGEN
> The *SYSGEN category that is owned by the specified category system is searched for the requested cartridge identifiers.

CNV The *CNV category that is owned by the specified category system is searched for the requested cartridge identifiers.

SHARE400
> The *SHARE400 category is searched for the requested cartridge identifiers. The category system field is not used for this category and must be set to blanks.

INSERT
> The *INSERT category is searched for the requested cartridge identifiers. The category system field is not used for this category and must be set to blanks.

EJECT
> The *EJECT category is searched for the requested cartridge identifiers. The category system field is not used for this category and must be set to blanks.

Category system
> CHAR(8). A specific system name or one of the following special values:

CURRENT
> All categories with the specified category name that are owned by the current system are searched for the requested cartridge indentifiers.

ALL All categories with the specifed category name that are owned by any system are searched for the requested cartridge identifiers.

Error code
> I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

RCTG0100 Format

The following table shows the information that is returned for the RCTG0100 format. For more details about the fields in the following table, see "Field Descriptions" on page 108.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Bytes returned

Offset			
Dec	Hex	Type	Field
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to cartridge information
12	C	BINARY(4)	Number of cartridge information entries
16	10	BINARY(4)	Length of cartridge information entry
		CHAR(*)	Cartridge information

Cartridge information

The current cartridge attributes. Each cartridge information entry has the following format. For more details about the fields in the following table, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	CHAR(6)	Cartridge ID
6	6	CHAR(6)	Volume ID
12	C	CHAR(10)	Tape library name
22	16	CHAR(10)	Category
32	20	CHAR(8)	Category system
40	28	CHAR(10)	Density
50	32	CHAR(7)	Change date
57	39	CHAR(6)	Change time
63	3F	CHAR(7)	Reference date
70	46	CHAR(6)	Reference time
76	4C	CHAR(10)	Location
86	56	CHAR(1)	Location indicator
87	57	CHAR(2)	Volume status
89	59	CHAR(17)	Owner ID
106	6A	CHAR(1)	Write protection
107	6B	CHAR(1)	Code
108	6C	CHAR(1)	Cartridge ID source
109	6D	CHAR(1)	Import/Export slot
110	6E	CHAR(2)	Media type

Field Descriptions

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Cartridge ID. The cartridge identifier for the cartridge.

Cartridge ID source. The type of cartridge ID. The following values can be returned:

0	The volume identifier is used for the cartridge ID.
1	The cartridge ID is generated by the system.
2	The cartridge ID was read from the bar code label on the cartridge.

Category. The category assigned to the cartridge.

Category system. The name of the system that owns the category assigned to the cartridge.

Change date. The date the cartridge was last used for output operations. The date is returned with the following format - CYYMMDD. If no change date is available the field is set to blanks.

Change time. The time the cartridge was last used for output operations. The time is returned with the following format HHMMSS. If no change time is available the field is set to blanks.

Code. The encoding used for the data on tape. The following values can be returned:

blank	The encoding is unknown.
0	ASCII encoding is used.
1	EBCDIC encoding is used.

Density. The last known cartridge density. If the density is unknown the field is set to blanks.

Import/Export slot. Whether or not the cartridge is in an Import/Export slot. The following values can be returned:

| 0 | The cartridge is not in an import/export slot. |
| 1 | The cartridge is in an import/export slot. |

Length of cartridge information entry. When cartridge information is available this field is set to the length, in bytes, of a single cartridge information entry. A value of zero is returned if the cartridge information is not available.

Location. The current location of the cartridge. The location will be set to either a slot number (if available) or a tape resource name. The following value may also be returned:

| *blanks* | Slot number is not available. |

Location indicator. This field indicates what type of location information is returned. Possible values follow:

| 0 | The cartridge is mounted in a tape resource in the tape library. |
| 1 | The cartridge is in a storage slot in the tape library. |

Media type. The media type read from the bar code label on the cartridge. If there is no bar code reader the field is set to blank.

Number of cartridge information entries. The number cartridge information entries returned. A value of zero is returned if the cartridge information is not available.

Offset to cartridge information. The offset, in bytes, to the cartridge information. A value of zero is returned if the cartridge information is not available.

Owner ID. The owner of the cartridge. This information is read from the tape volume labels. The following other values can be returned:

*NL	The cartridge is a non-labeled tape.
*BLANK	The owner ID in the tape labels was blank.
blanks	The owner ID is unknown

Reference date. The date the cartridge was last used. The date is returned with the following format - CYYMMDD. If no reference date is available the field is set to blanks.

Reference time. The time the cartridge was last used. The time is returned with the following format HHMMSS. If no reference time is available the field is set to blanks.

Tape Library name. The name of the tape library containing the cartridge.

Volume ID. The volume identifier for the cartridge read from the tape volume labels. The following other values can be returned:

*NL	The volume is a non-labeled tape.
blanks	The volume ID is unknown

Volume status. The status of the cartridge. Possible values follow:

01	The cartridge is available.
02	The cartridge is mounted.
03	The cartridge is not available.
04	The cartridge is ejected.
05	The cartridge is in error state.
06	The cartridge is inserted.
07	The cartridge has a duplicate cartridge ID.

Write Protection. Indicates whether or not the cartridge is write protected. The following values can be returned:

0	The cartridge is not write protected.
1	The cartridge is write protected.
blank	The cartridge write protect status is unknown.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF6708 E	Command ended due to error.
CPF6718 E	Cannot allocate device &1.

Message ID	Error Message Text
CPF673A E	Device &1 not varied on.
CPF6745 E	Device &1 not a media library device.
CPF67A1 E	Cartridge &1 not found.
CPF67A6 E	Category does not exist.
CPF67D2 E	Cartridge does not exist in specified category.
CPF9802 E	Not authorized to object &2 in &3.
CPF9814 E	Device &1 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R3

Retrieve Category List (QTARCGYL) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

The Retrieve Category List (QTARCGYL) API retrieves a list of the categories currently defined on the system.

Authorities and Locks

None

Required Parameter Group

Receiver variable
    OUTPUT; CHAR(*)

    The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
    INPUT; BINARY(4)

    The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
    INPUT; CHAR(8)

    The content and format of the information being returned.

    The RCGY0100 format must be used for the category list information. See "RCGY0100 Format" on page 112 to view the information returned for this format.

Error code
    I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

RCGY0100 Format

The following table shows the information that is returned for the RCGY0100 format. For more details about the fields in the following table, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to category list
12	C	BINARY(4)	Number of category list entries
16	10	BINARY(4)	Length of category list entry
		CHAR(*)	Category list

Category list

The list of categories that are currently defined for the system. Each category entry has the following format. For more details about the fields in the following table, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	CHAR(10)	Category Name
10	A	CHAR(8)	Category System

Field Descriptions

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Category name. The name of the defined category.

Category system. The name of the system owning the defined category. This field is blank for categories that are not owned by a specific system.

Length of category list entry. This field is set to the length, in bytes, of a single category list entry. A value of zero is returned if the category list is not available.

Number of category list entries. The number of category list entries returned. A value of zero is returned if the category list is not available.

Offset to category list. The offset, in bytes, to the category list. A value of zero is returned if the category list is not available.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF6708 E	Command ended due to error.
CPF67E4 E	Library device function not successful.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R3

Retrieve Device Capabilities (QTARDCAP) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Device description or resource name	Input	Char(10)
5	Device description or resource indicator	Input	Char(10)
6	Resources to list	Input	Char(1)
7	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Retrieve Device Capabilities (QTARDCAP) API retrieves information that is associated with a specified tape device description or tape resource name. The resource that is specified or associated with the specified device description must currently exist on the system.

The QTARDCAP API currently supports the following device types:
- Tape (TAP) devices
- Tape media library (TAPMLB) devices

Authorities and Locks

Device Description Authority
>> *USE «

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
INPUT; CHAR(8)

The content and format of the information being returned. The TAPE0100 format must be used for the tape device capabilities information. See "TAPE0100 Format" to view the information returned for this format.

Device description or resource name
INPUT; CHAR(10)

The name of the device description or resource for which the data is returned.

Device description or resource indicator
INPUT; CHAR(10)

Whether the name specified is a device description or a resource name. The possible values follow:

*DEVD	The name specified is a device description.
*RSRC	The name specified is a resource name.

Resources to list
INPUT; CHAR(1)

Which media library device resources are listed if the device description indicator in the device description or resource indicator parameter is *DEVD. This parameter is blank if the device requested is not a media library device description. If a media library device description is specified, the possible values are as follows:

1	All resources are listed (*ALL).
2	Allocated resources are listed (*ALLOCATED).
3	Unprotected resources are listed (*UNPROTECTED).
4	Deallocated resources are listed (*DEALLOCATED).
5	Both allocated and unprotected resources are listed (*AVAILABLE).
blank	The requested input device or resource is not a media library device, not a device description, or the information is not requested.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

TAPE0100 Format

The following table shows the information that is returned for the TAPE0100 format. For more details about the fields in the following table, see "Field Descriptions" on page 117.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Device description or resource name
18	12	CHAR(10)	Device description or resource indicator

Offset		Type	Field
Dec	Hex		
28	1C	CHAR(10)	Device resource name
38	26	CHAR(3)	Reserved
41	29	CHAR(1)	Device relationship to media library device
42	2A	CHAR(1)	Type of media library device
43	2B	CHAR(1)	Self-configured tape device
44	2C	CHAR(4)	Device type
48	30	CHAR(4)	Device model
52	34	BINARY(4)	Maximum block size
56	38	CHAR(1)	Logical-block-ID capability
57	39	CHAR(1)	Assign capability
58	3A	CHAR(1)	Overwrite capability
59	3B	CHAR(1)	Read-backward capability
60	3C	CHAR(1)	Cartridge-checking capability
61	3D	CHAR(1)	Device class
62	3E	CHAR(1)	Hardware data compression capability
63	3F	CHAR(1)	Label compaction supported
64	40	BINARY(4)	Offset to supported write densities
68	44	BINARY(4)	Number of supported write densities
72	48	BINARY(4)	Offset to supported read densities
76	4C	BINARY(4)	Number of supported read densities
80	50	BINARY(4)	Offset to supported improved-data-recording-capability (IDRC) densities
84	54	BINARY(4)	Number of supported improved-data-recording-capability (IDRC) densities
88	58	BINARY(4)	Optimum block size
92	5C	CHAR(1)	Space to end of data
93	5D	CHAR(1)	Space backward allowed
94	5E	CHAR(1)	Media library device with bar-code scanner
95	5F	CHAR(1)	Improved-data-recording-capability (IDRC) supported
96	60	CHAR(1)	Automatic cartridge mechanism supported
97	61	CHAR(2)	Bit mapping of all supported improved-data-recording-capability (IDRC) densities
99	63	CHAR(2)	Bit mapping of all supported write densities
101	65	CHAR(2)	Bit mapping of all supported read densities
103	67	CHAR(2)	Bit mapping of the highest supported density
105	69	CHAR(1)	Reserved
106	6A	CHAR(2)	Bit mapping of all supported character densities
108	6C	BINARY(4)	Offset to supported character densities
112	70	BINARY(4)	Number of supported character densities
116	74	BINARY(4)	Offset to resource list
120	78	BINARY(4)	Number of resources in resource list

Offset		Type	Field
Dec	Hex		
124	7C	BINARY(4)	Offset to resource status list
128	80	BINARY(4)	Number of resource status list entries
132	84	CHAR(1)	Reserved
133	85	CHAR(1)	QTACTLDV API supported
134	86	CHAR(1)	Media library device with door
≫ 135	87	CHAR(1)	Write-once-read-many (WORM) support
136	88	CHAR(1)	Virtual device
137	89	CHAR(3)	Reserved ≪
140	8C	BINARY(4)	Offset to density capability entries
144	90	BINARY(4)	Number of density capability entries
148	94	BINARY(4)	Length of density capability entries
152	98	BINARY(4)	Offset to system feature codes
156	9C	BINARY(4)	Length of system feature codes
160	A0	BINARY(4)	Acceptable read error thresholds
164	A4	BINARY(4)	Acceptable write error thresholds
168	A8	BINARY(4)	Instantaneous performance
172	AC	BINARY(4)	Offset to slot and station information
176	B0	BINARY(4)	Offset to device text
180	B4	BINARY(4)	Length of device text
≫ 184	B8	BINARY(4)	Offset to supported write-once-read-many (WORM) densities
188	BC	BINARY(4)	Number of supported write-once-read-many (WORM) densities ≪
		ARRAY(*) of CHAR(10)	Supported improved-data-recording-capability (IDRC) densities
		ARRAY(*) of CHAR(10)	Supported write densities
		ARRAY(*) of CHAR(10)	Supported read densities
		ARRAY(*) of CHAR(10)	Supported character densities
		ARRAY(*) of CHAR(10)	Resource list
		ARRAY(*) of CHAR(15)	Resource status list
		ARRAY(*) of CHAR(*)	Density capability entries
		CHAR(*)	System feature codes
		CHAR(*)	Device text
		CHAR(*)	Slot and station information
≫		ARRAY(*) of CHAR(10)	Supported write-once-read-many (WORM) densities ≪

Field Descriptions

Acceptable read error thresholds. The average minimum number of kilobytes (1KB = 1000 bytes) read between recovered read errors for the media to be considered acceptable. This information is to determine the reliability of a volume based on the number of errors that the device is reporting against that tape volume. This field is zero for devices that do not report this information.

Acceptable write error thresholds. The average minimum number of kilobytes (1KB = 1000 bytes) written between recovered write errors for the media to be considered acceptable. This information is to determine the reliability of a volume based on the number of errors that the device is reporting against that tape volume. This field is zero for devices that do not report this information.

Automatic cartridge mechanism supported. Whether the device has an automatic cartridge mechanism (loader or facility) known as an ACL (that is, 3490 type stand alone devices typically have an ACL) or an ACF (that is, 3590 B11 devices have an ACF). Possible values follow:

0   The device does not have an automatic cartridge mechanism.
1   The device does have an automatic cartridge mechanism.
blank   This field is not valid for a media library device.

Assign capability. Whether the specified device or resource has the capability to assign (reserve) a device to the system. This concept ensures that no other system can use the device because the system could not be successfully assigned (reserved) from the device. This feature fully enables devices that can be shared. Possible values follow:

0   The system cannot assign the device.
1   The system can assign the device.
blank   This is not an available value for a media library device.

Bit mapping of all supported character densities. A bit-mapped encoding of densities that correspond to the supported character densities. This field is not available for a media library device. Bit mappings are defined on the device level and may not match the following examples.

The following are the supported i5/OS densities and their corresponding bit-map representations for each device class. Note that self-configured tape devices do not follow these i5/OS bit-map representations but are defined in the device specifications. Values for the device class of 1/4-inch cartridge technology follow:

Hexadecimal Bit Map	Density or Format
'4000'x	*QIC24 (8000 bpi)
'2000'x	*QIC120 (10000 bpi)
'0800'x	*QIC525 (16000 bpi)
'0400'x	*QIC1000
'0200'x	*QIC2GB
'0100'x	*QIC3040
'0080'x	*QIC5010

Values for the device class of 1/2-inch cartridge technology follow:

Hexadecimal Bit Map	Density or Format
'8000'x	*FMT3480 (38000 bpi)
'4000'x	*FMT3490E
'2000'x	*FMT3590

Hexadecimal
Bit Map   **Density or Format**
'0800'x   *FMT3570
'0400'x   *FMT3570E

Values for the device class of 1/2-inch reel technology follow:

Hexadecimal
Bit Map   **Density or Format**
'8000'x   1600 bpi
'4000'x   3200 bpi
'2000'x   6250 bpi

Values for the device class of 8-mm cartridge technology follow:

Hexadecimal
Bit Map   **Density or Format**
'8000'x   *FMT2GB
'4000'x   *FMT5GB
'2000'x   *FMT7GB

Bit mapping of highest supported output density. A bit-mapped encoding of the highest supported output density on the device. This field is not available for a media library device. The definition of example bit maps can be found in the bit mapping of all supported character densities field.

Bit mapping of all supported improved-data-recording-capability (IDRC) densities. A bit-mapped encoding of densities that correspond to the supported IDRC densities. This field is not available for a media library device. The definition of example bit maps can be found in the bit mapping of all supported character densities field.

Bit mapping of all supported read densities. A bit-mapped encoding of densities that correspond to the supported read densities. This field is not available for a media library device. The definition of example bit maps can be found in the bit mapping of all supported character densities field.

Bit mapping of all supported write densities. A bit-mapped encoding of densities that correspond to the supported write densities. This field is not available for a media library device. The definition of example bit maps can be found in the bit mapping of all supported character densities field.

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Cartridge-checking capability. Whether the device communicates valid cartridge densities or formats. This capability allows i5/OS to verify cartridge density or formats. Devices that do not support this capability cannot send intelligent messages about the cartridge. When the device is unable to write to the cartridge, a generic error message usually results. It could be as simple as a cartridge not supported by the device. Possible values follow:

0   No device or resource that is requested has special cartridge-density checking capabilities.
1   The device requested has special cartridge-density checking capabilities.
blank This is not an available value for a media library device.

Device class. The class of the device. Possible values follow:

1   1/2-inch reel technology
2   1/2-inch cartridge technology
3   1/4-inch cartridge technology
4   8-mm technology
blank None of the above technologies or media library device

Density capability entries. The list of density capabilities based on density, format, or both. Each entry consists of the following format and has a length that is specified in the length of density capability entry field.

CHAR(10)   Density or format
CHAR(2)   Bit map representation of density
BINARY(4)   Instantaneous performance in megabytes per second for this density or format. One MB per second is returned for each field if the performance number is not reported by the device.
BINARY(4)   Maximum block size for this density or format.
BINARY(4)   Optimum block size for this density or format.

This format is repeated once for each density or format that is supported by the device. The number of entries in the array is specified in the number of density capability entries field. This field is not available for a media library device.

Device description or resource indicator. Whether the information is retrieved for a device description or a device resource name. This field is left-justified. Possible values follow:

*DEVD   The name specified is a device description.
*RSRC   The name specified is a device resource name.

Device description or resource name. The device description or resource name of the device.

Device relationship to media library device. The relationship of the device or resource to the media library device. Possible values follow:

0   The device is not a media library resource.
1   The device is a tape resource in a media library resource.
2   The device is a media library resource.

Device model. The device model number.

Device resource name. The resource name if the device information is for a device description.

Device text. The specific text that is reported by the device.

Device type. The device type number.

Hardware data compression capability. Whether hardware data compression (HDC) is supported. This field corresponds to the data compression (DTACPR) parameter on save commands. Possible values follow:

0   No device or resource that is requested has hardware data compression capabilities.
1   The device that is requested has hardware data compression capabilities.
blank This is not an available value for a media library device.

Improved-data-recording-capability (IDRC) supported. Whether the device supports IDRC (or compaction) at any density. If IDRC is supported, see the supported IDRC densities field. Possible values follow:

0	No device or resource that is requested has IDRC capabilities.
1	The device that is requested has IDRC capabilities.
blank	This is not an available value for a media library device.

Instantaneous performance. The highest instantaneous performance that is reported by the device. This value can be obtained from the density capability entries list on a density, format, or both. This value is the highest performance number that is specified in the density capability entries list. The value is in megabytes per second. If the device does not report this value, a value of 1 MB per second will be used. This is not an available value for a media library device and is set to zero for media library devices.

Label compaction supported. Whether the device volumes are written with compacted labels. Possible values follow:

0	The device does not generate labels with compaction.
1	The device generates labels with compaction if compaction is requested.

Length of density capability entry. The length, in bytes, of each entry in the density capability entries list. This field should be used in stepping through the array of density capability entries. Additional fields for each density or format in the density capability entries list may be added at any time.

Length of device text. The length, in bytes, of text that is reported by the device. If the device does not report specific text about the device, zero is returned.

Length of system feature codes. The length, in bytes, of the system feature codes entry. This field should be used in determining the system feature codes (type and model) of the device. If the device does not report this information, zero is returned.

Logical-block-ID capability. Whether the device allows fast access capabilities. This allows logical-block-identifier support through a tape management system that is registered with the registration facility and used in combination with the Media and Storage Extensions feature of i5/OS. Possible values follow:

0	No device or resource that is requested has fast access by logical-block-identifier capabilities.
1	The device that is requested has fast access by logical-block-identifier capabilities.
blank	This is not an available value for a media library device.

Maximum block size. The highest maximum block size that is supported by the device. If you use the maximum block size, tapes may be created that are not compatible with other device types. A tape that is created with a maximum block size can only be duplicated with the Duplicate Tape (DUPTAP) command to devices that support the same block size. This field is not available for a media library device and is set to zero for media library devices.

The maximum block size that a device supports may vary with density. Use the Density capability entries to determine the maximum block size for a specific density.

Media library device with bar-code scanner. Whether the device has a bar-code scanner, which is for a media library device. Possible values follow:

0	The device has no bar-code scanner.
1	The device has a bar-code scanner.

blank   This field is only valid for a media library device.

Media library device with door. Whether the media library device has a door that can be opened. Possible values follow:

0   The device does not have a door.
1   The device has a door.
blank   This field is only valid for a media library device.

Number of density capability entries. The number of density capability entries in the density capability entries list. This field is not available for a media library device and is set to zero when the input device is a media library device.

Number of resources in resource list. The number of resources that are listed in the resource list. This number is set to zero when the input device is not a media library device description (*DEVD).

Number of resource status list entries. The number of resource status entries for the resource status list. This number is set to zero when the input device is not a media library device description (*DEVD).

Number of supported improved-data-recording-capability (IDRC) densities. The number of supported IDRC (or compaction) densities that are specified in the supported IDRC densities field. This field is not available for a media library device and is set to zero when the input device is a media library device.

Number of supported character densities. The number of supported character densities that are specified in the supported character densities field. This field is not available for a media library device and is set to zero when the input device is a media library device.

Number of supported read densities. The number of supported read densities that are specified in the supported read densities field. This field is not available for a media library device and is set to zero when the input device is a media library device.

Number of supported write densities. The number of supported write densities that are specified in the supported write densities field. This field is not available for a media library device and is set to zero when the input device is a media library device.

» **Number of supported write-once-read-many (WORM) densities.** The number of supported WORM densities that are returned in the supported WORM densities array. This field is not available for a media library device and is set to zero when the input device is a media library device. This field will be set to sero if the device does not report WORM density information. «

Offset to density capability entries. The offset, in bytes, to the density capability entries list.

Offset to device text. The offset, in bytes, to the text that is reported by the device.

Offset to resource list. The offset, in bytes, to the resource list.

Offset to resource status list. The offset, in bytes, to the resource status list field.

Offset to slot and station information. The offset, in bytes, to the slot and station information field.

Offset to supported character densities. The offset, in bytes, to the supported character densities field.

Offset to supported improved-data-recording-capabilities (IDRC) densities. The offset, in bytes, to the supported improved-data-recording-capabilities (IDRC) densities field.

Offset to supported read densities. The offset, in bytes, to the supported read densities field.

Offset to supported write densities. The offset, in bytes, to the supported write densities field.

≫ Offset to supported write-once-read-many (WORM) densities. The offset, in bytes, to the supported WORM densities array. ≪

Offset to system feature codes. The offset, in bytes, to the system feature codes (types and models) of the device.

Optimum block size. The highest optimum block size supported by the device.

When USEOPTBLK(*YES) is specified as a parameter for a save operation, performance may be improved. The USEOPTBLK(*YES) parameter may cause tapes to be created that are not compatible with other device types. That is, a tape that is created with an optimum block size can only be duplicated with the Duplicate Tape (DUPTAP) command to devices that support the same block size. This field is not available for a media library device and is set to zero for media library devices.

The optimum block size that a device supports may vary with density. Use the Density capability entries to determine the optimum block size for a specific density.

Overwrite capability. Whether the device or resource that is specified has overwriting capabilities. This capability refers to being able to write a data file over an existing data file sequence on a tape volume. Some technologies only allow appending data to the end of tape or writing files to the beginning of tape. Possible values follow:

0	No device or resource that is requested has overwrite capabilities.
1	The device that is requested has overwrite capabilities.
blank	This is not an available value for a media library device.

QTACTLDV API supported. Whether the QTACTLDV API is supported for the device or resource that is specified. Possible values follow:

0	QTACTLDV API is not supported.
1	QTACTLDV API is supported.

Read-backward capability. Whether the device or resource that is specified has read-backward capabilities. Possible values follow:

0	No device or resource that is requested has read-backward capabilities.
1	The device that is requested has read-backward capabilities.
blank	This is not an available value for a media library device.

Reserved. An ignored field.

Resource list. The list of tape resources in the media library device.

Note: The data is an array of 10-byte entries. Each entry consists of a 10-byte density that is left-justified and padded with blanks. The number of entries in the array is specified in the number of resources in resource list field. The resource list is not valid when the input device is not a media library device description (*DEVD).

Resource status list. The list of statuses that correspond to resources in the resource list.

Note: The data is an array of 15-byte entries. Each entry consists of a 15-byte status that is left-justified and padded with blanks. The number of entries in the array is specified in the number of resource status list entries field. The resource status list is not valid when the input device is not a media library device description (*DEVD). Possible values follow:

*ALLOCATED   The tape resource is allocated to the media library device. The system has assigned the device.
*UNPROTECTED The tape resource is considered unprotected to the media library device. It is available to the media library device, but the system has not assigned the device. The system attempts to assign the resource when the resource is used.
*DEALLOCATED The tape resource is deallocated to the media library device. The resource is not available to the media library device, and the system has not assigned the device.

Self-configured tape device. Whether this resource or device is a self-configured tape device. Possible values follow:

0   The device is not a self-configured tape device.
1   The device is a self-configured tape device.

Slot and station information. This information is available only for a media library device. The information is returned in the following format:

BINARY(4)   Total number of storage slots
BINARY(4)   Total number of import/export stations

The above numbers are set to zero when the input device is not a media library device description (*DEVD) or when the information is not available.

For a media library device that requires a communication interface, the slot and station information is available only after the media library device description has been varied on.

Space backward allowed. Whether the device supports spacing backward or requires rewinding the device and respacing to the correct file. Possible values follow:

0   No device or resource that is requested has space-backward capabilities.
1   The device that is requested has space-backward capabilities.
blank This is not an available value for a media library device.

Space to end of data. The function that allows quick access to the end of the logical tape when a command specifies SEQNBR(*END), such as the Save Library (SAVLIB) command. This function allows a significant improvement on performance when the tape is positioned at the beginning of the tape and needs to be positioned a long way into the tape. If the device does not support this function, i5/OS spaces by the file marks until the end of data is reached. Possible values follow:

0   No device or resource that is requested has space to end of data capabilities.
1   The device that is requested has space to end of data capabilities.
blank This is not an available value for a media library device.

Supported improved-data-recording-capability (IDRC) densities. The list of densities that support IDRC (or compaction) on this device.

Note: The data is an array of 10-byte entries. Each entry consists of a 10-byte density that is left-justified and padded with blanks. The number of entries in the array is specified in the number of supported improved-data-recording-capability (IDRC) densities field. This field is not available for a media library device.

System feature codes. The system feature codes (type and model) that the device reports. The first 4 bytes of the system feature codes field displays which feature codes are available or reported by the device. The 32 bits of the 4 bytes are defined as follows:

Bit 0   iSeries default feature code provided
Bit 1-31   Not used

Each bit that is identified in the 32 bits as defined above is represented with 4 bytes of device type and model in hexadecimal representation. These 4 bytes are repeated for each bit that is turned on in the first 4 bytes of the system feature codes field. The length of this field determines how much information is returned from the device and is contained in the length of system feature codes. The length of this field includes the first 4 bytes of bit definitions. That is, if only bit 0 is on, then the length of the field is 8 bytes, which includes the 4 bytes of bit definition and the 4 bytes of the one system feature code.

Supported character densities. The list of densities that are supported on this device.

Note: The data is an array of 10-byte entries. Each entry consists of a 10-byte density that is left-justified and padded with blanks. The number of entries in the array is specified in the number of supported character densities field. This field is not available for a media library device.

Supported read densities. The list of valid read densities that are supported by this device.

Note: The data is an array of 10-byte entries. Each entry consists of a 10-byte density that is left-justified and padded with blanks. The number of entries in the array is specified in the number of supported read densities field. This field is not available for a media library device.

Supported write densities. The list of valid write densities that are supported by this device.

Note: The data is an array of 10-byte entries. Each entry consists of a 10-byte density that is left-justified and padded with blanks. The number of entries in the array is specified in the number of supported write densities field. This field is not available for a media library device.

» **Supported write-once-read-many (WORM) densities.** The list of valid WORM densities that are supported by this device.

Note: The data is an array of 10-byte entries. Each entry consists of a 10-byte density that is left-justified and padded with blanks. The number of entries in the array is specified in the number of supported WORM densities field. This data is not available for a media library device. «

Type of media library device. The media library technology. Possible values follow:

0   The device is not a media library device.
1   Library commands are communicated to the library robotic device.
2   Library commands are communicated to a communications line.
3   Library commands are communicated to the library robotic device and the device.

» **Virtual device.** Whether or not the device is a virtual device. Possible values follow:

0   The device is not a virtual device.
1   The device is a virtual device. «

》Write-once-read-many (WORM) support. Whether the device supports WORM at any density. If WORM is supported, see the supported WORM densities array. Possible values follow:

0   No device or resource that is requested has WORM capabilities.
1   The device that is requested has WORM capabilities.
blank Indicates this field is not applicable for a media library device. 《

Error Messages

Message ID	Error Message Text
CPF3C90 E	Literal value cannot be changed.
CPF6707 E	Format name &1 not valid.
CPF6708 E	Command ended due to error.
CPF6709 E	Parameter &3 not correct.
CPF6710 E	Specified length not correct.
CPF6721 E	Device &1 not a tape device.
CPF672F E	Resource &1 not found.
CPF673F E	Device &1 does not support &2.
CPF9814 E	Device &1 not found.
CPF9825 E	Not authorized to device &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

Retrieve Device Information (QTARDINF) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Device	Input	Char(10)
4	Format name	Input	Char(8)
5	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Retrieve Device Information (QTARDINF) API retrieves information that is associated with a specified device description.

The QTARDINF API currently supports the following device types:
- Tape (TAP) devices
- Tape media library (TAPMLB) devices

Specifically, it retrieves information about the current status and mode of the specified device. The device must be varied on at run time of the API.

Authorities and Locks

Device Description Authority
>> *USE <<

Required Parameter Group

Receiver variable
> OUTPUT; CHAR(*)

> The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
> INPUT; BINARY(4)

> The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Device
> INPUT; CHAR(10)

> The name of the device description for which the data is to be returned.

Format name
> INPUT; CHAR(8)

> The content and format of the information being returned. The TADS0100 format must be used for the retrieve device information. See "TADS0100 Format" to view the information returned for this format.

Error code
> I/O; CHAR(*)

> The structure in which to return error information. For the format of the structure, see Error Code Parameter.

TADS0100 Format

The following table shows the information that is returned for the TADS0100 format. For more details about the fields in the following table, see "Field Descriptions."

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Number of active jobs

Field Descriptions

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Number of active jobs. The number of active jobs for the device. For stand-alone devices the value returned will be zero or one depending on whether it is in use at the current time. Note that since stand-alone tape devices require an exclusive lock through system functions, the device will only allow one user at a time. For media library devices, multiple users can access the tape media library at the same time through shared locks; therefore, any number of jobs may be active at the same time. This field is designed to allow users to better utilize one resource tape media library such as the 3570 tape media library device.

Error Messages

Message ID	Error Message Text
CPF3C90 E	Literal value cannot be changed.
CPF6707 E	Format name &1 not valid.
CPF6708 E	Command ended due to error.
CPF6709 E	Parameter &3 not correct.
CPF6710 E	Specified length not correct.
CPF6721 E	Device &1 not a tape device.
CPF672F E	Resource &1 not found.
CPF673A E	Device &3 not varied on.
CPF673F E	Device &1 does not support &2.
CPF9814 E	Device &1 not found.
CPF9825 E	Not authorized to device &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R1

Retrieve Device Status (QTARDSTS) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Device description	Input	Char(10)
5	Resource name	Input	Char(10)
6	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Retrieve Device Status (QTARDSTS) API retrieves dynamic status information for the specified device and for any currently mounted tape cartridge. The device description must be varied on. The resource that is associated with a specified tape media library device description must currently exist on the system.

Note: If the device status has been changed by a manual operation or by another system sharing the device, the information will not be accurate.

The QTARDSTS API currently supports the following device types:
- Tape (TAP) devices
- Tape media library (TAPMLB) devices

Authorities and Locks

Device Description Authority
　　　　*USE

Required Parameter Group

Receiver variable
　　　　OUTPUT; CHAR(*)

　　　　The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
　　　　INPUT; BINARY(4)

　　　　The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
　　　　INPUT; CHAR(8)

　　　　The content and format of the information being returned.

　　　　The RDST0100 format must be used for the tape device status information. See "RDST0100 Format" to view the information returned for this format.

Device description
　　　　INPUT; CHAR(10)

　　　　The name of the device description for which the data is returned.

Resource name
　　　　INPUT; CHAR(10)

　　　　When the Device description parameter specifies a tape media library device description, this parameter can be used to specify the resource name of a tape device within the tape media library for which data is returned. This parameter must be set to blanks when only the tape media library information is needed, or when the device description is for a tape device.

Error code
　　　　I/O; CHAR(*)

　　　　The structure in which to return error information. For the format of the structure, see Error code parameter.

RDST0100 Format

The following table shows the information that is returned for the RDST0100 format. For more details about the fields in the following table, see "Field Descriptions" on page 131.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to current cartridge information
12	C	BINARY(4)	Number of current cartridge information entries
16	10	BINARY(4)	Length of current cartridge information entry

Offset			
Dec	Hex	Type	Field
20	14	BINARY(4)	Offset to device information
24	18	BINARY(4)	Number of device information entries
28	1C	BINARY(4)	Length of device information entry
32	20	BINARY(4)	Offset to label information
36	24	BINARY(4)	Number of label information entries
40	28	BINARY(4)	Length of label information entry
44	2C	BINARY(4)	Offset to position information
48	30	BINARY(4)	Number of position information entries
52	34	BINARY(4)	Length of position information entry
56	38	BINARY(4)	Offset to tape media library information
60	3C	BINARY(4)	Number of tape media library information entries
64	40	BINARY(4)	Length of tape media library information entry
		CHAR(*)	Current cartridge information
		CHAR(*)	Device information
		CHAR(*)	Label information
		CHAR(*)	Position information
		CHAR(*)	Tape media library information

Current cartridge information

The following table shows the current cartridge information that is returned. This information is only available when a command was previously issued to use the device and one or more of the following conditions are present:

- There is an open tape file for the device.
- The device is in leave processing.
- The device is varied on with assign(*YES).
- > » The device is a virtual tape device and an Image catalog is loaded.«

This information is not returned when the specified device description is a tape media library and no device resource is provided. The information returned may not be accurate if there was an error reported during the previously issued command. For more details about the fields in the following table, see "Field Descriptions" on page 131.

Offset			
Dec	Hex	Type	Field
0	0	CHAR(6)	Volume ID
6	6	CHAR(6)	Cartridge ID
12	C	CHAR(10)	Current cartridge density
22	16	CHAR(1)	Write protected
» 23	17	CHAR(10)	Image catalog name
33	21	CHAR(6)	Mounted volume ID «

Device information

The following table shows the device information that is returned. This information is only available when there is an active job using the tape device or the specified device description is a tape media library and a device resource is provided that is owned by the tape media library. For more details about the fields in the following table, see "Field Descriptions" on page 131.

Offset			
Dec	Hex	Type	Field
0	0	CHAR(10)	Current command
10	A	CHAR(26)	Job using the device
36	24	CHAR(1)	Category is mounted
37	25	CHAR(10)	Mounted category name
47	2F	CHAR(8)	Mounted category system
55	37	CHAR(6)	Mounted cartridge ID
61	3D	CHAR(1)	Cartridge order for category
62	3E	CHAR(10)	Target category
72	48	CHAR(8)	Target system
80	50	CHAR(10)	Mount identifier

Label information

The following table shows the label information that is returned. This information is only available when there is an open tape file for the tape device or the device is in leave processing. This information is not returned when the specified device description is a tape media library and no device resource is provided. For more details about the fields in the following table, see "Field Descriptions" on page 131.

Offset			
Dec	Hex	Type	Field
0	0	CHAR(1)	Label type
1	1	CHAR(1)	Tape encoding
2	2	CHAR(10)	Current file sequence number
12	C	CHAR(10)	Current multi-volume sequence number
22	16	CHAR(80)	Volume label
102	66	CHAR(80)	Last processed HDR1/TRL1 label
182	B6	CHAR(80)	Last processed HDR2/TRL2 label

Position information

The following table shows the position information that is returned. This information is only available when there is an open tape file for the tape device or the device is in leave processing. This information is not returned when the specified device description is a tape media library and no device resource is provided. For more details about the fields in the following table, see "Field Descriptions" on page 131.

Offset			
Dec	Hex	Type	Field
0	0	CHAR(1)	In leave processing

Offset			
Dec	Hex	Type	Field
1	1	CHAR(1)	At logical EOT
2	2	CHAR(1)	At logical BOT
3	3	CHAR(1)	Reserved
4	4	BINARY(4)	Current tape mark count

Tape media library information

The following table shows the tape media library information that is returned. This information is only available when the specified device description is a tape media library and a command was previously issued to use the tape media library. For more details about the fields in the following table, see "Field Descriptions."

Note: For a 3494 tape media library device the tape media library information is only updated when a DSPTAPSTS command is issued.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Number of used slots
4	4	BINARY(4)	Number of available slots
8	8	BINARY(4)	Number of high capacity slots
12	C	BINARY(4)	Number of import/export stations

Field Descriptions

At logical BOT. Whether the media is currently positioned at logical BOT. For standard labeled format tapes this indicator is set whenever the tape is positioned within the first label group. For non-labeled format tapes this indicator is set whenever the tape is positioned within the first file on tape. Possible values follow:

0   The media is not currently positioned at logical BOT.
1   The media is currently positioned at logical BOT.

At logical EOT. Whether the media is currently positioned at logical EOT. Possible values follow:

0   The media is not currently positioned at logical EOT.
1   The media is currently positioned at logical EOT.

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Cartridge ID. The Cartridge identifier for the currently mounted media. This field will be blank if the tape does not have a bar code label or if it is not in a tape media library with a bar code reader.

Cartridge order for category. When there is a category mounted on the specified resource within a tape media library this field shows the cartridge order. Possible values follow:

0	The cartridges will be mounted in next available (*NEXTAVAIL) order.
1	The cartridges will be mounted in sequential (*SEQ) order.
blank	There is no category mounted.

Category is mounted. Whether there currently is a category mounted on the device. Possible values follow:

0	There is not a category mounted on the device.
1	There is a category mounted on the device.
blank	The device is not allocated to a tape media library.

Current cartridge density. The density of the currently mounted media.

Current command. The current command executed against the tape device. If the specific command is not known, one of the following values is returned:

CPP CMD	A CL command was executed.
SAVE CMD	A save command was executed.
RST CMD	A restore command was executed.
COMMAND	An unknown type of command was executed.

Current file sequence number. The current sequence number of the file being processed. This field will be blanks if no files have been processed yet.

Current multi-volume sequence number. The current multi-volume sequence number for the mounted media. This field will be blank if no media is mounted or if the tape is a non-labeled tape.

Current tape mark count. The current number of tape marks away from BOT.

≫**Image catalog name.** The name of the image catalog loaded on the virtual tape device. When there is no image catalog loaded, the field will be set to blanks. ≪

In leave processing. Whether the last completed tape command used an ending option of *LEAVE. Possible values follow:

0	The media is not currently left in position.
1	The media is currently left in position.

Job using the device. The qualified job name of the job currently using the device. This field will be all blanks if there is no active job using the device.

Label type. The label format of the currently mounted tape. Possible values follow:

0	The media uses standard label format.
1	The media uses non-labeled format.

Last processed HDR1/TRL1 label. The last processed HDR1 or TRL1 label on the media. This field will be all blanks for a Non-labeled tape, or if no labels have been processed yet.

Last processed HDR2/TRL2 label. The last processed HDR2 or TRL2 label on the media. This field will be all blanks for a Non-labeled tape or if no labels have been processed yet.

Length of current cartridge information entry. When current cartridge information is available this field is set to the length, in bytes, of a single current cartridge information entry. A value of zero is returned if the current cartridge information is not available.

Length of device information entry. When device information is available this field is set to the length, in bytes, of a single device information entry. A value of zero is returned if the device information is not available.

Length of label information entry. When label information is available this field is set to the length, in bytes, of a single label information entry. A value of zero is returned if the label information is not available.

Length of position information entry. When position information is available this field is set to the length, in bytes, of a single position information entry. A value of zero is returned if the position information is not available.

Length of tape media library information entry. When tape media library information is available this field is set to the length, in bytes, of a single tape media library information entry. A value of zero is returned if the tape media library information is not available.

Mounted cartridge ID. For a resource within a tape media library device the cartridge identifier for the cartridge that was last mounted is returned. When the specified device description is a tape device, the field is set to blanks.

Mounted category name. The name of the category mounted on the specified resource within a tape media library. When there is no category mounted, the field is set to blanks.

Mounted category system. The name of the system owning the category mounted on the specified resource within a tape media library. When there is no category mounted or the category is not owned by a specific system, the field is set to blanks.

Mount identifier. The mount identifier assigned to the category mounted on the specified resource within a tape media library. When there is no mount identifier assigned the value *NONE is returned. When there is no category mounted, the field is set to blanks.

≫ **Mounted volume ID.** The name of the volume which is currently mounted on the virtual tape device. When there is no image catalog loaded or no volume mounted, the field is set to blanks. ≪

Number of available slots. The number of empty storage slots in the tape media library.

Number of current cartridge information entries. If current cartridge information is available, a value of one is returned. A value of zero is returned if the current cartridge information is not available.

Number of device information entries. If device information is available, a value of one is returned. A value of zero is returned if the device information is not available.

Number of high capacity slots. The number of high capacity storage slots in the tape media library.

Number of import/export stations. The number of import/export stations for the tape media library.

Number of label information entries. If label information is available, a value of one is returned. A value of zero is returned if the label information is not available.

Number of position information entries. If position information is available, a value of one is returned. A value of zero is returned if the position information is not available.

Number of tape media library information entries. If tape media library information is available, a value of one is returned. A value of zero is returned if the tape media library information is not available.

Number of used slots. The number of storage slots in the tape media library that are currently being used.

Offset to current cartridge information. The offset, in bytes, to the current cartridge information. A value of zero is returned if the current cartridge information is not available.

Offset to device information. The offset, in bytes, to the current job information. A value of zero is returned if the device information is not available.

Offset to label information. The offset, in bytes, to the label information. A value of zero is returned if the label information is not available.

Offset to position information. The offset, in bytes, to the position information. A value of zero is returned if the position information is not available.

Offset to tape media library information. The offset, in bytes, to the tape media library information. A value of zero is returned if the tape media library information is not available.

Tape Encoding. The encoding scheme being used for the mounted media. Possible values follow:

0	ASCII format tape.
1	EBCDIC format tape.

Target category. The category that the cartridges will be changed to after they are used. When there is no category mounted, the field is set to blanks.

Target system. The name of the system owning the target category. When there is no category mounted, the field is set to blanks.

Volume ID. The Volume identifier for the currently mounted media. This field will be blank for non-labeled tapes or if the tape cannot be read.

Volume label. The volume label for the currently mounted media. This field will be blank for a non-labeled tape or if the tape cannot be read.

Write protected. Whether the mounted media is write protected. Possible values follow:

0	The media is not write protected.
1	The media is write protected.
≫ *blank*	Write protect status is unknown. ≪

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C3C E	Value for parameter &1 is not valid.
CPF3C90 E	Literal value cannot be changed.

Message ID	Error Message Text
CPF3CF1 E	Error code parameter not valid.
CPF6708 E	Command ended due to error.
CPF6721 E	Device &1 not a tape device.
CPF672F E	Resource &1 not found.
CPF67B0 E	Tape resource &2 not in specified library device.
CPF9802 E	Not authorized to object &2 in &3.
CPF9814 E	Device &1 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R2

Retrieve Job Media Library Attributes (QTARJMA) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Qualified job name	Input	Char(26)
5	Internal job identifier	Input	Char(16)
6	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

The Retrieve Job Media Library Attributes (QTARJMA) API retrieves the specified job's current settings for the media library attributes. For more information, see Tape information in the Storage Solutions topic.

Authorities and Locks

Job Authority
> *JOBCTL, if the job for which information is retrieved has a different user profile from that of the job that calls the QTARJMA API.

Required Parameter Group

Receiver variable
> OUTPUT; CHAR(*)

> The variable that is to receive the information requested. You can specify the size of an area smaller than the format requested as long as you specify the receiver length parameter correctly. As a result, the API returns only the data the area can hold.

Length of receiver variable
> INPUT; BINARY(4)

> The length of the receiver variable. The length must be at least 8 bytes. If the variable is not long enough to hold the information, the data is truncated. If the length is larger than the size of the receiver variable, the results are not predictable.

Format name
> INPUT; CHAR(8)

> The format name RJMA0100 is the only valid format name used by this API. For more information, see "RJMA0100 Format" on page 136.

Qualified job name
> INPUT; CHAR(26)

> The name of the job for which information is to be returned. The qualified job name has three parts:

Job name
> CHAR(10). A specific job name or the following special value:

*	The job that this program is running in. The rest of the qualified job name parameter must be blank.
*INT	The internal job identifier locates the job. The user name and job number must be blank.

User name
> CHAR(10). A specific user profile name, or blanks when the job name is a special value or *INT.

Job number
> CHAR(6). A specific job number, or blanks when the job name specified is a special value or *INT.

Internal job identifier
> INPUT; CHAR(16)

> The internal identifier for the job. The List Job (QUSLJOB) API creates this identifier. If you do not specify *INT for the job name parameter, this parameter must contain blanks. With this parameter, the system can locate the job more quickly than with a job name.

Error code
> I/O; CHAR(*)

> The structure in which to return error information. For the format of the structure, see Error Code Parameter.

RJMA0100 Format

The following table lists the fields for the receiver variable in the RJMA0100 format. For more information about each field, see "Field Descriptions" on page 137.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to list of device entries
12	C	BINARY(4)	Number of device entries
16	10	BINARY(4)	Length of a device entry
20	14	CHAR(12)	Reserved

Offset		Type	Field
Dec	**Hex**		
Offsets vary. These fields repeat in the order listed, for each media library device that has attributes defined.		CHAR(10)	Media library device
		CHAR(6)	Reserved
		BINARY(4)	Resource allocation priority
		BINARY(4)	Wait time for initial mount
		BINARY(4)	Wait time for end of volume mount
		CHAR(4)	Reserved

Field Descriptions

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Length of a device entry. The length, in bytes, of a device entry. A value of zero is returned if the list is empty.

Media library device. The name of the media library device that the attributes apply to. The special value supported is:

DEFAULT   The attributes apply to all media libraries that do not have attributes defined.

Number of device entries. The number of entries in the device list returned for this format. A value of zero is returned if the list is empty.

Offset to the list of device entries. The offset, in bytes, to the list of device entries returned with this format. A value of zero is returned if the list is empty.

Reserved. All reserved fields will contain hexadecimal zeros.

Resource allocation priority. The priority that the specified job will be given when the job requests a tape resource within a media library device.

Exceptions:
- Value of -2 implies *DEV. The priority specified in the device description will be used when the job requests a tape resource.
- Value of -31 implies *JOB. The specified job's run-time priority will be used for the resource allocation priority when the job requests a tape resource.

Wait time for end of volume mount. The maximum amount of time, in minutes, a request will wait for the allocation of a tape resource to mount the next volume after the end of volume is reached.

Exceptions:
- Value of -2 implies *DEV. The end of volume mount wait time specified in the device description will be used.
- Value of -8 implies *NOMAX. The specified job will wait until a resource becomes available.
- Value of -31 implies *JOB. The specified job's default wait time will be used to calculate the wait time. The time is calculated by rounding the default wait time, in seconds, to the next highest minute.

- Value of -32 implies *IMMED. The specified job will not wait for a resource to become available.

Wait time for initial mount. The maximum amount of time, in minutes, a request will wait for the allocation of a tape resource to mount the first volume.

Exceptions:
- Value of -2 implies *DEV. The initial mount wait time specified in the device description will be used.
- Value of -8 implies *NOMAX. The specified job will wait until a resource becomes available.
- Value of -31 implies *JOB. The specified job's default wait time will be used to calculate the wait time. The time is calculated by rounding the default wait time, in seconds, to the next highest minute.
- Value of -32 implies *IMMED. The specified job will not wait for a resource to become available.

Error Messages

Message ID	Error Message Text
CPF1343 E	Job &3/&2/&1 not valid job type for function.
CPF136A E	Job &3/&2/&1 not active.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C51 E	Internal job identifier not valid.
CPF3C52 E	Internal job identifier no longer valid.
CPF3C53 E	Job &3/&2/&1 not found.
CPF3C54 E	Job &3/&2/&1 currently not available.
CPF3C55 E	Job &3/&2/&1 does not exist.
CPF3C58 E	Job name specified is not valid.
CPF3C59 E	Internal identifier is not blanks and job name is not *INT.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF6708 E	Command ended due to error.
CPF67B6 E	&3/&2/&1 not authorized to do requested operation.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R3

Retrieve Media Definition (QSRRTVMD, QsrRetrieveMediaDefinition) API

Required Parameter Group:

1	Qualified media definition name	Input	Char(20)
2	Receiver variable	Output	Char(*)
3	Length of receiver variable	Input	Binary(4)
4	Format name	Input	Char(8)
5	Error code	I/O	Char(*)

Service Program Name: QSRLIB01
Default Public Authority: *USE
Threadsafe: No

The Retrieve Media Definition (OPM, QSRRTVMD; ILE, QsrRetrieveMediaDefinition) API retrieves a media definition specified by the user. ≫ A **media definition** defines the devices, media, and data format

to be used in parallel by a save or restore operation. For more information about using a media definition, see Save to multiple devices in the Backup and recovery topic. «

Authorities and Locks

Media Definition Authority
　　　*USE

Library Authority
　　　*EXECUTE

Media Definition Lock
　　　*SHRNUP

Required Parameter Group

Qualified media definition name
　　　INPUT; CHAR(20)

　　　The media definition to be retrieved. The first 10 characters contain the media definition name. The second 10 characters contain the name of the library in which the media definition is located.

　　　You can use the following special values for the library name. It should be noted, however, that the library name that is actually used is not passed back to the user. Care should be taken when using these special values to avoid unexpected results.

CURLIB	The job's current library is used to locate the media definition. If no library is specified as the current library for the job, the QGPL library is used.
LIBL	The library list is used to locate the media definition.

Receiver variable
　　　OUTPUT; CHAR(*)

　　　The variable that is to hold all the information defining the use of multiple tape files for a save or restore operation. See "Format of Receiver Variable" on page 140 for the format of the information.

Length of receiver variable
　　　INPUT; BINARY(4)

　　　The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
　　　INPUT; CHAR(8)

　　　The name of the format for the receiver variable. The valid values are:

»*MDFN0100*	Format name that was used to create the media definition «
TAPE0100	Tape devices and media
»*TAPE0200*	Tape devices and media (extended) «

　　　» Media definitions created with format TAPE0100 can be retrieved with format TAPE0200. The additional options in format TAPE0200 will be set to their default values.

　　　Media definitions created with format TAPE0200 can be retrieved with format TAPE0100. The additional options in format TAPE0200 will be lost. «

Error code
　　　I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Format of Receiver Variable

» For format MDFN0100, the retrieved data consists of the following structure. For detailed descriptions of the fields, see "Field Descriptions for Receiver Variable."

Format MDFN0100

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(8)	Format name

For other formats, the retrieved data consists of a header and a set of device definitions and media file definitions. « The following defines the format for the header. For detailed descriptions of the fields, see "Field Descriptions for Receiver Variable."

Format TAPE0100

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Maximum parallel device resources
12	C	BINARY(4)	Minimum parallel device resources
16	10	BINARY(4)	Offset to first device definition
20	14	BINARY(4)	Number of device definitions
		CHAR(*)	Device definitions

Format TAPE0200

Offset			
Dec	Hex	Type	Field
0	0		All fixed fields from format TAPE0100
24	18	BINARY(4)	Length of header
28	1C	BINARY(4)	Device allocation
32	20	BINARY(4)	Save format
		CHAR(*)	Device definitions

«

Field Descriptions for Receiver Variable

Bytes available. The number of bytes available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned. If this is less than the bytes available, the information returned is not complete.

» Device allocation. When to allocate the tape devices. The default value is 0. The possible values are:

0   All tape devices are allocated at the beginning of the operation.

1   One tape device is allocated at the beginning of a save operation. Additional devices are allocated when data is ready to be written, at which time the number of devices specified for the Minimum parallel device resources field is required.

2   The number of devices specified for the Minimum parallel device resources field is allocated at the beginning of a save operation. Additional devices are allocated when data is ready to be written.

«

Device definitions. A description of the devices to be used. See "Device Definition Format" for the format of a device definition.

» Format name. The name of the format that was used to create the media definition.

Length of header. The length of the fixed portion of the header information. «

Maximum parallel device resources. The maximum number of device resources to use in parallel. The possible values are 0 through 32. If 0 is specified, the value assumed is the total number of media file definitions specified in all of the device definitions.

Minimum parallel device resources. The minimum number of device resources to use in parallel. A save or restore operation will end if fewer resources are available. A restore operation will also end if any of the devices specified have no resources available. The possible values are 0 through 32. If 0 is specified, the value assumed is the number of device definitions specified.

Number of device definitions. The number of device definitions for the media definition. The possible values are 1 through 32.

Offset to first device definition. The offset from the beginning of the receiver variable to the first device definition for the media definition.

» Save format. Whether to save data in serial format or parallel format. This field is ignored for restore operations. The default value is -2. The possible values are:

-2   If one library is saved, it is saved in parallel format. If more than one library is saved, all libraries are saved in serial format.

-1   All data is saved in serial format.

0   All data is saved in parallel format.

«

Device Definition Format

Format TAPE0100

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Offset to next device definition
4	4	CHAR(10)	Device name
14	E	CHAR(2)	Reserved

Offset			
Dec	Hex	Type	Field
16	10	BINARY(4)	Offset to first media file definition
20	14	BINARY(4)	Number of media file definitions
		CHAR(*)	Media file definitions

Format TAPE0200

Offset			
Dec	Hex	Type	Field
0	0		All fixed fields from format TAPE0100
24	18	BINARY(4)	Length of device definition
		CHAR(*)	Media file definitions

《

Field Descriptions for Device Definition

Device name. The name of a tape device description or tape media library device description.

》**Length of device definition.** The length of the fixed portion of the device definition. 《

Media file definitions. A description of the media files to be used on this device. See "Media File Definition Format" for the format of a media file definition.

Number of media file definitions. The number of media file definitions for the device.

Offset to first media file definition. The offset from the beginning of the receiver variable to the first media file definition for the device.

Offset to next device definition. The offset from the beginning of the receiver variable to the next device definition for the media definition.

Reserved. An ignored field.

Media File Definition Format

Format TAPE0100

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Offset to next media file definition
4	4	BINARY(4)	Sequence number
8	8	BINARY(4)	Offset to volume identifier array
12	C	BINARY(4)	Number of volume identifiers
16	10	BINARY(4)	Length of volume identifier
20	14	BINARY(4)	Starting volume array element
		CHAR(*)	Volume identifier array

Format TAPE0200

Offset			
Dec	Hex	Type	Field
0	0		All fixed fields from format TAPE0100
24	18	BINARY(4)	Length of media file definition
		CHAR(*)	Volume identifier array

≪

Field Descriptions for Media File Definition

≫ **Length of media file definition.** The length of the fixed portion of the media file definition. ≪

Length of volume identifier. The number of bytes in each volume identifier.

Number of volume identifiers. The number of volume identifiers used for the tape file. The possible values are 0 through 75. If 0 is specified, the volume currently placed in the device is used. If 0 is specified for a tape media library device, volume identifiers must be supplied by using the Tape Management exit program during the save or restore operation.

Offset to next media file definition. The offset from the beginning of the receiver variable to the next media file definition for the device.

Offset to volume identifier array. The offset from the beginning of the receiver variable to the first volume identifier for the media file.

Sequence number. The tape file sequence number for a tape media file.

The possible values are:

0	A save operation begins after the last sequence number on the starting volume. A restore operation searches the starting volume for a media file containing any of the objects to restore.
1-16777215	The sequence number of the tape file.

Starting volume array element. The element in the volume identifier array containing the volume on which the save or restore operation should begin. The possible values are 0 through the number of volume identifiers.

Volume identifier array. An array of volume identifiers. The save or restore operation will use the volumes in the order specified, beginning with the starting volume array element. If additional volumes are needed after the last array element is used, the save or restore operation will call the Tape Management exit program or prompt the user to provide each additional volume. The possible value for a volume identifier is:

Volume identifier   The identifier of a volume.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.

Message ID	Error Message Text
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF9800 E	All CPF98xx messages could be signaled. xx is from 01 to FF.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V4R4

Retrieve Tape Labels (QTARTLBL) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Device description	Input	Char(10)
5	Request qualifiers	Input	Char(*)
6	Length of request qualifiers	Input	Binary(4)
7	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Retrieve Tape Labels (QTARTLBL) API retrieves the label information for the files on a tape cartridge. This API provides a function similar to the Display Tape (DSPTAP) command.

Authorities and Locks

Device Description Authority
    *USE

Required Parameter Group

Receiver variable
    OUTPUT; CHAR(*)

    The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
    INPUT; BINARY(4)

    The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
    INPUT; CHAR(8)

    The content and format of the information being returned.

The RLBL0100 format must be used for the label information. See "RLBL0100 Format" to view the information returned for this format.

Device description
    INPUT; CHAR(10)

The name of the tape device or tape library device containing the cartridge for which the label information is to be returned.

Request qualifiers
    INPUT; CHAR(*)

The request qualifiers specify which cartridge to use, what labels to return, and what action to perform after the label information is retrieved. The following table shows the information that can be specified. Default values are used for any fields that are not specified. For more details about the fields in the following table see "Field Descriptions" on page 146.

Offset			
Dec	Hex	Type	Field
0	0	CHAR(6)	Requested volume identifier
6	6	CHAR(17)	Requested file label
23	17	CHAR(10)	Starting file sequence number
33	21	CHAR(10)	Ending file sequence number
43	2B	CHAR(1)	End option
44	2C	CHAR(*)	Reserved

Length of request qualifiers
    INPUT; BINARY(4)

The length, in bytes, of the Request qualifiers provided. The following values are allowed:

0   No request qualifiers are provided.
6   Only the requested volume identifier is specified.
23   The requested volume identifier and the requested file label is specified.
33   The requested volume identifier, requested file label, and starting file sequence number is is specified.
43   The requested volume identifier, requested file label, starting file sequence number, and ending file sequence number is specified.
44 or greater   All request qualifiers are specified.

Error code
    I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

RLBL0100 Format

The following table shows the information that is returned for the RLBL0100 format. For more details about the fields in the following table, see "Field Descriptions" on page 146.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Device name

Offset			
Dec	Hex	Type	Field
18	12	CHAR(6)	Cartridge ID
24	18	CHAR(80)	Volume Label
104	68	CHAR(1)	Code
105	69	CHAR(1)	Standard label
106	6A	CHAR(1)	Leading tape mark
107	6B	CHAR(10)	Density
117	75	CHAR(2)	Reserved
119	77	CHAR(1)	Additional label entries
120	78	BINARY(4)	Offset to label information
124	7C	BINARY(4)	Number of label information entries
128	80	BINARY(4)	Length of label information entry
		CHAR(*)	Label information

Label information

The label information. Each label information entry has the following format. For more details about the fields in the following table, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	CHAR(80)	Label 1
80	50	CHAR(80)	Label 2
160	A0	CHAR(32)	Logical block ID
192	C0	CHAR(10)	Sequence number
202	CA	CHAR(10)	Multi-volume sequence number
212	D4	CHAR(8)	S/36 File type

For more information regarding IBM Standard Volume Labels and Data Set Labels 1 and 2, select the appropriate topic below:

- IBM Standard Volume Label (VOL1)
- IBM Standard Data Set Label 1 (HDR1/EOV1/EOF1)
- IBM Standard Data Set Label 2 (HDR2/EOV2/EOF2)

Field Descriptions

Additional label entries. Indicates if there were labels on tape that were not returned because the return data would have exceeded 16,000,000 Bytes. The following values can be returned:

0   No additional label entries.
1   There were more label entries to return.

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Cartridge ID. The cartridge identifier for the media. This field will be blanks if the device is not a tape library device.

Code. The encoding used for the data on tape. The following values can be returned:

0	ASCII encoding is used.
1	EBCDIC encoding is used.
blank	Non-labeled tape.

Density. The density that is used for the data on the tape.

Device name. The tape device or tape library device that was used.

Ending file sequence number. The ending sequence number for which the labels are to be retrieved. When a numeric value is specified it must be left justified and padded with blanks on the right. The ending sequence number can not be less than the starting file sequence number. The following special values are supported:

ONLY	Only the file specified by the starting file sequence number will be retrieved, if it matches the specified file label. **This is the default value.**
LAST	The labels on tape that match the specifed file label are to be retrieved, starting with the file specified by the Starting file sequence number and ending with the last file sequence number on tape.
blank	The ending file sequence number will not be used. This value is only valid when the special value *ALL is specified for the starting file sequence number.

End option. The operation to perform on the tape volume after the retrieve completes. The following special values are supported:

0	Rewind the tape. **This is the default value.**
1	Rewind and unload the tape.
2	Leave the tape at the current position.

Label 1. The contents of the first trailer label for the file sequence.

Label 2. The contents of the second trailer label for the file sequence.

Leading tape mark. Indicates if the tape was a leading tape mark tape. The following values can be returned:

0	Not a leading tape mark tape.
1	Leading tape mark tape.

Length of label information entry. When label information is available this field is set to the length, in bytes, of a single label information entry. A value of zero is returned if the tape is not a standard labeled tape, if there are no labels on the tape, or if the receiver variable is not large enough to hold any label information entries.

Logical block ID. The logical block identifier for the file sequence.

Multi-volume sequence number. The sequence number for the tape volume.

Number of label information entries. If there is room in the receiver variable for 1 or more complete label information entries, this value is set to the number of complete label information entries returned. The Bytes returned value can be used to determine if any partial label information entries were returned.

If there is room in the receiver variable for only part of the first label information entry, this value is set to 1 and it will be necessary to use the Bytes returned value to determine which label information fields are valid.

A value of zero is returned if the tape is not a standard labeled tape, if there are no labels on the tape, or if the receiver variable is not large enough to hold any label information entries.

Note: A maximum of 72,726 entries can be returned.

Offset to label information. The offset, in bytes, to the label information. A value of zero is returned if the tape is not a standard labeled tape, if there are no labels on the tape, or if the receiver variable is not large enough to hold any label information.

Requested file label. The data file label that is to be retrieved. The data file label is a maximum of 17 characters in length and is case sensitive. The following special value is supported:

*ALL   All file labels within the specified sequence number range are to be retrieved. **This is the default value.**

Note: When a specific file label is requested, the label information will only be returned for the first file found that matches the requested file label.

Requested volume identifier. The tape volume for which the labels are to be retrieved. For a tape library device this field should be set to a cartridge identifier to be mounted and used. The following special value is supported:

blanks   The currently mounted tape cartridge is used. **This is the default value.**

Reserved. An ignored field. This field is set to hexadecimal zeros for output fields and must be set to hexadecimal zeros for input fields.

S/36 File type. The System/36 file type for the file sequence. The following file types can be returned:
- ARCHIVE
- APARFILE
- COPYFILE
- EXCHANGE
- LIBRFILE
- SAVEFLDR
- SAVELIBR

Sequence number. The file sequence number.

Standard label. Indicates if the tape was a standard labeled tape. The following values can be returned:

0   Non-labeled tape.

1   Standard labeled tape.

Starting file sequence number. The starting sequence number for which the labels are to be retrieved. When a numeric value is specified it must be left justified and padded with blanks on the right. The following special values are supported:

*ALL   All labels on tape are to be retrieved that match the specified file label.

             Note: When this value is used the ending file sequence number must be set to blanks.
*FIRST   The labels on tape that match the specified file label are to be retrieved, starting with the first file label. **This is the default value.**

Volume label. The volume label read from the tape. This field is set to blanks for a non-labeled tape.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3C3C E	Value for parameter &1 is not valid.
CPF3C4B E	Value not valid for field &1.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF4108 E	Media error on volume &8 device &4.
CPF4119 E	Device &4 cannot process loaded volume.
CPF4120 E	Device &4 or an attached gateway device reported a hardware failure.
CPF4121 E	Interface error while using device &4.
CPF412C E	Cartridge &6 not found.
CPF412D E	Cartridge not available.
CPF412F E	Cartridge &6 not available.
CPF414C E	Command not allowed.
CPF414E E	Library device &5 not ready.
CPF414F E	Library device storage slots full.
CPF415A E	*MOUNTED not correct.
CPF415E E	Resource request timed out on device &4.
CPF416A E	No device available.
≫ CPF41B3 E	No more volumes to mount from catalog. ≪
≫ CPF41B4 E	Virtual tape volume not available. ≪
≫ CPF41B5 E	Virtual tape volume not found. ≪
CPF6708 E	Command ended due to error.
CPF6718 E	Cannot allocate device &1.
CPF6720 E	Incorrect volume &2 found on device &1.
CPF6721 E	Device &1 not a tape device.
CPF6723 E	File not found on volume &2 on device &1.
CPF6724 E	File label &5 not found on volume &2.
CPF6725 E	Ending file sequence number less than starting sequence number.
CPF673A E	Device &3 not varied on.
CPF6751 E	Load failure occurred on device &4.
CPF6760 E	Device &1 not ready.
CPF6772 E	Volume on device &1 cannot be processed.
CPF67E6 E	Volume &2 not correct.

Message ID	Error Message Text
CPF9802 E	Not authorized to object &2 in &3.
CPF9814 E	Device &1 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R3

Save Object (QsrSave) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Error code	I/O	Char(*)

Service Program Name: QSRLIB01
Default Public Authority: *USE
Threadsafe: No

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

For detailed restrictions on using this API to save objects in libraries or to save document library objects, see Save file systems in the Backup and recovery topic.

Authorities and Locks

User Space

> *User Space Authority*
> > *USE

> *User Space Library Authority*
> > *EXECUTE

> *User Space Lock*
> > *EXCLRD

Objects to Be Saved, Devices, Save While Active, Save-While-Active Message Queue, and Output

> *Locking*
> > See Object locking rules in the Backup and recovery topic for information on object locking for the Save Object (SAV) command.

> *Authority*
> > In the iSeries Security Reference book, see the Appendix about authorities required for the Save Object (SAV) command.

Required Parameter Group

Qualified user space name
> INPUT; CHAR(20)

> The user space that is to hold all the information for the save operation. The first 10 characters

contain the user space name. The second 10 characters contain the name of the library where the user space is located. See "User Space Format" for the format of the information in the user space.

You can use the following special values for the library name. However, it should be noted that the library name that is actually used is not passed back to the user. Care should be taken when you use these special values to avoid unexpected results.

*CURLIB   The job's current library is used to locate the user space. If no library is specified as the current library for the job, the QGPL library is used.

*LIBL   The library list is used to locate the user space.

Error code
    I/O; CHAR(*)

    The structure in which to return error information. For the format of the structure, see Error Code Parameter.

User Space Format

The following defines the format for the information in the user space. For detailed descriptions of the fields in the user space format, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Number of variable length records
4	4	BINARY(4)	Offset to first record
8	8	CHAR(8)	Reserved
Note: These fields repeat for each variable length record.			
		BINARY(4)	Key
		BINARY(4)	Offset to next record
		CHAR(8)	Reserved
		CHAR(*)	Data

If the length of the data is longer than the key identifier's data length, the data will be truncated at the right. No message will be issued.

If the specified data length is shorter than the key field's defined data length, an error message is returned for binary fields. If the field is a character field, the data is padded with blanks and an error message will not be returned.

Note: This does not apply to keys that allow a list of values to be specified. In these cases, the amount of data read is based on the specified number of entries in the list.

If keys are duplicated in the user space, only the last value for a given key is used for the save operation.

Each variable length record must be 4-byte aligned. If not, unpredictable results may occur.

Field Descriptions

Data. The data used to specify the value for the given key.

Key. The parameter of the Save Object (SAV) command to specify. See "Valid Keys" on page 152 for the list of valid keys.

Offset to first record. The offset from the beginning of the user space to the first variable length record.

Offset to next record. The offset from the beginning of the user space to the next variable length record.

Number of variable length records. The number of variable length records that are passed in the user space. The valid range is from 2 through 25.

Reserved. Reserved. The possible value is:

x'00'   This field should contain x'00's.

Valid Keys

The following table lists the valid keys for the key field area of the variable length record. For detailed descriptions of the keys, see the "Field Descriptions" on page 153.

Some messages for this API refer to parameters and values of the Save Object (SAV) command. This table can also be used to locate the key names that correspond to the SAV command parameters. The field descriptions contain, in addition to detailed descriptions, the corresponding parameter values.

The object path name key and the device path name key are required keys. The other keys are optional.

Key	Type	Field	SAV Command Parameter
1	CHAR(*)	Device path name	DEV
2	CHAR(*)	Object path name	OBJ
3	CHAR(1)	Directory subtree	SUBTREE
4	CHAR(1)	System	SYSTEM
5	CHAR(40)	Change period	CHGPERIOD
6	CHAR(1)	Object precheck	PRECHK
7	CHAR(10)	Target release	TGTRLS
8	CHAR(*)	Update history	UPDHST
9	CHAR(*)	Volume identifier	VOL
10	CHAR(*)	Label	LABEL
11	BINARY(4)	Sequence number	SEQNBR
12	CHAR(7)	Expiration date	EXPDATE
13	CHAR(1)	End of media option	ENDOPT
14	CHAR(1)	Clear	CLEAR
15	CHAR(1)	Data compression	DTACPR
16	CHAR(1)	Data compaction	COMPACT
17	CHAR(*)	Optical file	OPTFILE
18	CHAR(1)	Save while active	SAVACT
19	CHAR(*)	Save-while-active message queue	SAVACTMSGQ
20	CHAR(*)	Output	OUTPUT, INFTYPE
21	CHAR(1)	Use optimum block size	USEOPTBLK
22	CHAR(1)	Save-while-active option	SAVACTOPT
23	CHAR(10)	ASP device name	ASPDEV
24	CHAR(*)	Name pattern	PATTERN

Key	Type	Field	SAV Command Parameter
25	CHAR(2)	Scan objects	SCAN

Field Descriptions

The values shown in parentheses are the corresponding values for the SAV command parameters.

ASP device name. The names of the auxiliary storage pool (ASP) devices to be included in the save operation. The default is *ALLAVL. The possible values are:

*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 job has a linked ASP group, all independent ASPs in the linked ASP group.
*SYSBAS	The operation includes the system ASP and all basic user ASPs.
*ASPGRP	If the job has a linked ASP group, all independent ASPs in the linked ASP group are included in the save operation.
ASP device name	The operation includes the specified independent ASP.

Change period. A date and time range. Objects that changed within the range are saved.

If this key is not specified, the default of *ALL will be used for the start date and time and the end date and time.

Offset			
Dec	Hex	Type	Field
		CHAR(10)	Start date
		CHAR(10)	Start time
		CHAR(10)	End date
		CHAR(10)	End time

End date. The date before which objects that have changed are saved. The possible values are:

*ALL	No ending date is specified. All objects changed since the starting date are saved.
end-date	The date before which objects that have changed are saved in the format CYYMMDD:

C	Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
YY	Year
MM	Month
DD	Day

End time. The time on the end date before which objects that have changed are saved. The possible values are:

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

| *end-time* | The time on the end date before which objects that have changed are saved in the format HHMMSS: |

HH	Hour
MM	Minute
SS	Second

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

Start date. The date after which objects that have changed are saved. The possible values are:

| **ALL* | No starting date is specified. All objects changed prior to the ending date are saved. |
| **LASTSAVE* | Objects are saved that have changed since the last time they were saved with update history. |

Notes:

1. If this value is specified, the value *ALL must be specified on all other elements of this key.

2. For file systems that are accessed through the network server, the PC archive attribute is used. For other file systems, the system archive attribute is used.

| *start-date* | The date after which objects that have changed are saved in the format CYYMMDD: |

C	Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
YY	Year
MM	Month
DD	Day

Start time. The time on the start date after which objects that have changed are saved. The possible values are:

| **ALL* | All times of day are included in the range. |
| *start-time* | The time on the start date after which objects that have changed are saved in the format HHMMSS: |

HH	Hour
MM	Minute
SS	Second

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

Clear. Whether active data on the media is cleared or replaced automatically. 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 the tape. Replacing active data on optical media replaces only the optical files created by this operation. The default is 0.

Notes:

1. Clearing a tape does not initialize it. Before the save command is issued, you should initialize the tape to a standard label format by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.

2. Clearing an optical volume does initialize it.

3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

The possible values are:

0   None of the media is cleared automatically. If the save operation encounters active data on a tape 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. (*NONE)

1   All of the media is cleared automatically. (*ALL)

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

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

       Note: This value is not valid for save files.

3   Active data on the media is replaced automatically. Optical volumes are not initialized. Tapes and save files are cleared automatically in the same way as the value 1. (*REPLACE)

Data compaction. Whether device data compaction is performed. The default is 1. The possible values are:

0   Device data compaction is not performed. (*NO)

1   Device data compaction is performed if the data is saved to tape and all tape devices specified for the device key support the compaction feature. (*DEV)

       Note: If 1 is specified for the data compaction key and 2 is specified for the data compression key, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

       If 1 is specified for the data compaction key and 1 is specified for the data compression key, both device data compaction and device data compression are performed if supported on the device.

Data compression. Whether data compression is performed. 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. The default is 2. The possible values are:

0   No data compression is performed. (*NO)

1   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 or save file, software data compression is done. Low (SNA) software compression is used for all devices except optical DVD, which uses medium (TERSE) software compression. (*YES)

       Note: If 1 is specified for the data compression key and 1 is specified for the data compaction key, both device data compaction and device data compression are performed if supported on the device.

2   If the tape device has the hardware compression feature installed, processing proceeds as if 1 were specified for the data compression key. If the compression feature is not installed or if save data is written to optical or save file, processing proceeds as if 0 were specified for the data compression key. (*DEV)

       Note: If 2 is specified for the data compression key and 1 is specified for the data compaction key, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

3   If the save operation is to a save file or optical, low (SNA) 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. Low compression is usually faster than medium or high compression. The compressed data is usually larger than if medium or high compression is used. (*LOW)

4   If the save operation is to a save file or optical, medium (TERSE) 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. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used. (*MEDIUM)

5   If the save operation is to a save file or optical, high (LZ1) 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. High compression is usually slower than low and medium compression.The compressed data is usually smaller than if low or medium compression is used. (*HIGH)

Device path name. The path name of the device to which the objects are saved.

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Number in array
		BINARY(4)	Offset to first device path name
Note: These fields repeat for each device path name			
		BINARY(4)	Offset to next device path name
		CHAR(12)	Reserved
		CHAR(*)	Device path name

Device path name. The path name of the device to which the objects are saved. The path name should be specified in the Qlg_Path_Name_T format. If a pointer is specified in the path name format, it must be 16-byte aligned. If not, unpredictable results may occur. For more information on this structure, see Path name format. The possible value is:

device-path-name   The path name of the » media definition, « media library device, optical device, save file, or tape device used to save the objects. If a » media definition, « media library device, optical device, or save file path name is specified, it must be the only element in the array.

» For information about creating and using a media definition, see Save to multiple devices in the Backup and recovery topic and the "Create Media Definition (QSRCRTMD, QsrCreateMediaDefinition) API" on page 14 (QSRCRTMD, QsrCreateMediaDefinition) API. «

Number in array. The number of devices used during the save operation. The possible values are:

1-4   The number of devices used during the save operation.

Offset to first device path name. The offset from the beginning of the user space to the first device path name in the list. The possible value is:

n   The offset from the beginning of the user space to the first device path name in the list.

Offset to next device path name. The offset from the beginning of the user space to the next device path name in the list. The possible value is:

n   The offset from the beginning of the user space to the next device path name in the list. If the current device path name is the last device path name in the array, this value should be 0.

Reserved. Reserved. The possible value is:

x'00'   This field should contain x'00's.

Directory subtree. Whether the directory subtrees are included in the save operation. The default is 1.
The possible values are:

0   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. (*NONE)
1   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. (*ALL)
2   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.
           (*DIR)
3   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. (*OBJ)
4   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). (*STG)

End of media option. The operation that is performed automatically on the tape or optical volume after
the save operation ends. If more than one volume is used, this key applies only to the last volume used;
all other volumes are unloaded when the end of the volume is reached. The default is 0.

Note: This parameter is valid only if a tape or optical device name is specified. For optical devices, 2 is
the only value supported; 0 and 1 are ignored.

The possible values are:

0   The tape is automatically rewound, but not unloaded, after the operation ends. (*REWIND)
1   The tape does not rewind or unload after the operation ends. It remains at the current position on the tape
           drive. (*LEAVE)
2   The tape is automatically rewound and unloaded after the operation ends. Some optical devices eject the
           volume after the operation ends. (*UNLOAD)

Expiration date. The media in the device cannot be overwritten until the expiration date. The default is
0999999. The possible values are:

0999999   The media in the device is protected permanently. (*PERM)
date   The date when protection for the media ends in the format CYYMMDD:

                   C   Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.

                   YY   Year

                   MM   Month

                   DD   Day

Label. The file identifier of the media to be used for the save operation. The default is *GEN. The
possible values are as follows:

*GEN   The system generates the label.

       • 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) commands.

       • For objects in other file systems, the label is SAVyyyymmdd, where yyyymmdd is:

         yyyy   Year

         mm   Month

         dd   Day

file-identifier   The identifier (maximum of 17 characters) of the tape file used for the save operation.

Name pattern. Specifies a pattern to be used to include or omit objects.

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
		BINARY(4)	Offset to first pattern name
Note: These fields repeat for each pattern name.			
		BINARY(4)	Offset to next pattern name
		CHAR(1)	Option
		CHAR(11)	Reserved
		CHAR(*)	Pattern name

Number in array. The number of pattern names. The possible values are:

1-n   The number of pattern names.

Pattern Name. Specifies a pattern name. The possible value is:

pattern-name   The object name or pattern that can match many names.

Offset to first pattern name. The offset from the beginning of the user space to the first pattern name in the list. The possible value is:

n   The offset from the beginning of the user space to the first pattern name in the list.

Offset to next pattern name. The offset from the beginning of the user space to the next pattern name in the list. The possible value is:

n   The offset from the beginning of the user space to the next pattern name in the list. If the current pattern name is the last pattern name in the array, this value should be 0.

Option. Whether names that match the pattern should be included or omitted from the save operation.

Note: The subtree key specifies whether the subtrees are included or omitted.

The possible values are:

0   All objects which are included by the OBJ parameter are included in the save except those objects which match the PATTERN parameter. This value overrides objects that are included with option 1 and is intended to be used to omit a subset of a previously selected patterns. (*OMIT)

1   Only objects which are included by the OBJ parameter and match the PATTERN parameter are included in the save, unless overridden by an omit specification. (*INCLUDE)

Reserved. Reserved. The possible value is:

x'00'   This field should contain x'00's.

Object path name. The path name of the object to save. You can specify a pattern for this path name.

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
		BINARY(4)	Offset to first object path name
Note: These fields repeat for each object path name.			
		BINARY(4)	Offset to next object path name
		CHAR(4)	Reserved
		CHAR(1)	Option
		CHAR(7)	Reserved
		CHAR(*)	Object path name

Number in array. The number of object path names to be saved. The possible values are:

1-300   The number of object path names to be saved.

Object path name. The path name of the object to save. You can specify a pattern for this path name. The path name should be specified in the Qlg_Path_Name_T format. If a pointer is specified in the path name format, it must be 16-byte aligned. If not, unpredictable results may occur. For more information on this structure, see Path name format. The possible value is:

object-path-name   The object path name or pattern that can match many names.

Offset to first object path name. The offset from the beginning of the user space to the first object path name in the list. The possible value is:

n   The offset from the beginning of the user space to the first object path name in the list.

Offset to next object path name. The offset from the beginning of the user space to the next object path name in the list. The possible value is:

n   The offset from the beginning of the user space to the next object path name in the list. If the current object path name is the last object path name in the array, this value should be 0.

Option. Whether names that match the pattern should be included or omitted from the save operation. When determining whether the name matches a pattern, name patterns are always treated as relative to the current working directory.

Note: The subtree key specifies whether the subtrees are included or omitted.

The possible values are:

0   The objects that match the object name pattern are not saved. This value overrides objects that are included with option 1 and is intended to be used to omit a subset of a previously selected pattern. (*OMIT)

1   The objects that match the object name pattern are saved, unless overridden by an omit specification. (*INCLUDE)

Reserved. Reserved. The possible value is:

x'00'   This field should contain x'00's.

Object precheck. Whether the save operation ends if any of the selected objects cannot be saved. The default is 0. The possible values are:

0   The save operation does not end. Objects that can be saved are saved. (*NO)

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

Optical file. The path name of the optical file that is used for the save operation. The path name should be specified in the Qlg_Path_Name_T format. If a pointer is specified in the path name format, it must be 16-byte aligned. If not, unpredictable results may occur. For more information on this structure, see Path name format. The default is '*'. The possible values are:

'*'	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	The path name of the optical file that is used for the save operation, beginning with the root directory of the volume.

Output. 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.

Offset		Type	Field
Dec	Hex		
		CHAR(1)	Option
		CHAR(1)	Type of output information
		CHAR(14)	Reserved
		CHAR(*)	Output path name

Option. Whether a list of information about the saved objects is created. The default is 0. The possible values are:

0   No output is created. (*NONE)

1   The output is printed with the job's spooled output. (*PRINT)

2   The output is directed to an existing stream file or user space specified by the output path name.

Output path name. The path name of the existing stream file or user space to which the output of the API is directed. The path name should be specified in the Qlg_Path_Name_T format. If a pointer is specified in the path name format, it must be 16-byte aligned. If not, unpredictable results may occur. For more information on this structure, see Path name format. The possible value is:

path-name   The path name of the existing stream file or user space to which the output of the API is directed.

Reserved. Reserved. The possible value is:

x'00'   This field should contain x'00's.

Type of output information. The type of information that is directed to the spooled file, stream file, or user space specified for the output key. The possible values are:

0   The file contains information about the command, and an entry for each directory. (*SUMMARY)
1   The file contains information about the command, an entry for each directory, and an entry for each object that was not successfully saved. (*ERR)
2   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. (*ALL)

Save while active. Whether an object can be updated while it is being saved. The default is 0. The possible values are:

0   The objects that are in use are not saved. Objects cannot be updated while they are being saved. (*NO)
1   Objects can be saved and used at the same time. The object checkpoints can occur at different times. (*YES)
2   Objects can be saved and used at the same time. All of the object checkpoints occur at the same time. (*SYNC)

Save-while-active message queue. The path name of the message queue that the save operation uses to notify the user that save-while-active checkpoint processing is complete.

Offset			
Dec	Hex	Type	Field
		CHAR(1)	Option
		CHAR(15)	Reserved
		CHAR(*)	Save-while-active message-queue path name

Option. Whether a message should be used to notify the user that save-while-active checkpoint processing is complete. The default is 0. The possible values are:

0   No notification message is sent. (*NONE)
1   The notification message is sent to the work station message queue. (*WRKSTN)
2   The notification message is sent to the specified save-while-active message-queue path name.

Reserved.

       Reserved. The possible value is:

x'00'   This field should contain x'00's.

Save-while-active message-queue path name. The path name of the message queue that will be used to notify the user that save-while-active checkpoint processing is complete. The path name should be specified in the Qlg_Path_Name_T format. If a pointer is specified in the path name format, it must be 16-byte aligned. If not, unpredictable results may occur. For more information on this structure, see Path name format. The possible value is:

Save-while-active   The path name of the message queue.
message-queue
path name

Save-while-active option. The options that should be used with the save-while-active key. The possible values are:

0   No special save-while-active options will be used. (*NONE)
1   When 1 or 2 is specified for the save-while-active key, objects will be enabled to be saved when they are being updated if the corresponding system attribute for the object is set.

       This option should be used only by applications to save objects that are associated with the application and that have additional backup and recovery considerations. See Save your system while it is active in the Backup and Recovery topic for additional information.

Scan objects. Whether objects will be scanned while being saved when exit programs are registered with any of the integrated file system scan-related exit points and whether objects that previously failed a scan should be saved.

The integrated file system exit points are:
- Integrated File System Scan on Open Exit Program
- Integrated File System Scan on Close Exit Program

Offset			
Dec	**Hex**	**Type**	**Field**
		CHAR(1)	Scan during save
		CHAR(1)	Save failed objects

Scan during save. Whether objects will be scanned while being saved when exit programs are registered with any of the integrated file system scan-related exit points. The default is 0. The possible values are:

0   Objects will not be scanned by the scan-related exit programs. (*NO)
1   Objects will be scanned according to the rules described in the scan-related exit programs. (*YES)

Save failed objects. Whether objects that previously failed a scan should be saved. The default is 0. The possible values are:

0   Objects that have either previously failed a scan or that fail a scan by a QIBM_QP0L_SCAN_OPEN exit program during this save will not be saved. (*NOSAVFAILED)
1   Objects that have either previously failed a scan or that fail a scan during this save will be saved. (*SAVFAILED)

Sequence number. The tape file sequence number to be used. The default is -1. The possible values are:

-1   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. (*END)
1-16777215   The sequence number of the file. Any existing files on the tape at or beyond this sequence number are overwritten.

System. Whether to process objects that exist on the local system or remote systems. The default is 0. The possible values are:

0	Only local objects are processed. (*LCL)
1	Only remote objects are processed. (*RMT)
2	Both local and remote objects are processed. (*ALL)

Target release. The release level of the operating system on which you intend to use the object being saved. The default is *CURRENT. The possible values are:

*CURRENT	The object is to be restored to, and used on, the release of the operating system that is currently running on your system. The object can also be restored to a system with any subsequent release of the operating system.
*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	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.
	When you specify 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 and Recovery book for a complete list of valid values.

Update history. Whether to update the save history on objects saved with this save operation. The save history is used when *LASTSAVE is specified for the start time value of the change period key on a subsequent save operation. The possible values include:

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
Note: This field repeats for each update history value.			
		CHAR(1)	Update history

Number in array. The number of update history values. The possible values are:

1-2	The number of update history values.

Update history. Whether to update the save history on objects saved with this save operation. The save history is used when *LASTSAVE is specified for the start time value of the change period key on a subsequent save operation. The default is 0. The possible values include:

0	The save history is not updated. (*NO)
1	The save history is updated. For file systems that are accessed through the network server, the PC archive attribute is set to No. For other file systems, the system archive attribute is set to No. (*YES)
2	The system save history is updated. The system archive attribute is set to No. (*PC)
3	The PC save history is updated. The PC archive attribute is set to No. (*SYS)

Use optimum block size. Whether the optimum block size is used for the save operation. The default is 1. The possible values are:

0	The optimum block size supported by the device is not used. Save uses the default block size supported by all device types. The tape volume can be duplicated to any media format by using the Duplicate Tape (DUPTAP) command. (*NO)
1	The optimum block size supported by all devices is used. If the optimum block size is used, the following can occur:

- 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 that supports the same block size that was used.
- The value for the data compression key is ignored.

Volume identifier. The volume identifiers of the volumes, or the cartridge identifier of a tape in a tape media library device, on which data is saved. The volumes must be placed in the device in the order specified on this key. After all specified volumes are filled, the save operation continues on whatever volumes are mounted on the device.

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
		BINARY(4)	Length of each volume identifier
		BINARY(4)	Offset to first volume identifier
Note: These fields repeat for each volume identifier.			
		BINARY(4)	Offset to next volume identifier
		CHAR(*)	Volume identifier

Length of each volume identifier. The character length of each of the volume identifiers. The possible value follows:

n	The size of a single volume identifier. The maximum size of a tape volume identifier is 6 characters. The maximum size of an optical volume identifier is 32 characters. If a volume identifier larger than the maximum size is entered for this key, it is truncated to the maximum size.

Number in array. The number of volume identifiers that are used during the save operation. The default is 0. The possible values are:

0	The volume currently placed in the device is used. If 0 is specified for a tape media library device, volume identifiers must be supplied by using the Tape Management exit program during the save or restore operation. If 0 is specified, the length of each volume identifier value is ignored. (*MOUNTED)

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

1-75	The number of volume identifiers used during the save operation.

Offset to first volume identifier. The offset from the beginning of the user space to the first volume identifier in the list. The possible value is:

n	The offset from the beginning of the user space to the first volume identifier in the list.

Offset to next volume identifier. The offset from the beginning of the user space to the next object volume identifier in the list. The possible value is:

n　　　　　　　　The offset from the beginning of the user space to the next volume identifier in the list. If the current volume identifier is the last volume identifier in the array, this value should be 0.

Volume identifier. The volume identifiers of one or more volumes to be used. The possible value is:

Volume identifier　　The volume identifiers of one or more volumes to be used.

Dependencies between Keys

The following two tables list the dependencies between the different keys. If the dependency pertains only to a certain value, then that value is also shown (key = *n*, where *n* is the value). Otherwise, if the dependency is true for all values of the key, then only the name of the key is given.

The following table lists the conditions where specifying a certain key forces the use of another key.

If you specify...	...must be specified
Device = optical library device	Volume identifier

The following table lists the conditions where specifying a certain key excludes the user from using another key or a particular value of that key.

If you specify...	...cannot be specified
≫ Device = media definition	Optical file Sequence number Volume identifier ≪
Device = optical device	Label Sequence number Use optimum block size
Device = save file	Clear = 2 End of media option Expiration date Label Optical file Sequence number Use optimum block size Volume identifier
Device = tape device	Optical file
Save while active = 0	Save-while-active message queue Save-while-active option

Relationship to SAV Command

Because of the relationship between the QsrSave API and the SAV command, the following situations should be noted:

- Message text: Several messages produced by this API refer to parameters or values of the SAV command (for example, *AFTER). To determine which key a given parameter corresponds to, see "Valid Keys" on page 152. To determine which key value a given parameter value corresponds to, see "Field Descriptions" on page 153.

- Command type: The command type listed for the API on headings of displays and print files is QsrSave for integrated file system objects. If QsrSave is used to save objects in libraries or document

library objects (DLOs), the generated output indicates that the corresponding library or document library objects (DLO) command generated the media.

Error Messages

Message ID	Error Message Text
CPF0001 E	Error found on &1 command.
CPF24B4 E	Severe error while addressing parameter list.
CPF3700 E	All CPF37xx messages could be signalled. xx is from 01 to FF.
CPF3800 E	All CPF38xx messages could be signalled. xx is from 01 to FF.
CPF3C31 E	Object type &1 is not valid.
CPF3C4D E	Length &1 for key &2 not valid.
CPF3C81 E	Value for key &1 not valid.
CPF3C82 E	Key &1 not valid for API &2.
CPF3C83 E	Key &1 not allowed with value specified for key &2.
CPF3C84 E	Key &1 required with value specified for key &2.
CPF3C85 E	Value for key &1 not allowed with value for key &2.
CPF3C86 E	Required key &1 not specified.
CPF3C87 E	Key &1 allows one value with special value.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF5729 E	Not able to allocate object &1.
CPF9800 E	All CPF98xx messages could be signaled. xx is from 01 to FF.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V4R3

Save Object List (QSRSAVO) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Error Code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Save Object List (QSRSAVO) API saves a list of objects ≫ or spooled files ≪ specified by the user. The list of objects, as well as any additional information needed for the save operation, is generated by the user into a user space.

Authorities and Locks

User Space

> *User Space Authority*
> *USE
>
> *User Space Library Authority*
> *EXECUTE
>
> *User Space Lock*
> *SHRNUP

Objects to Be Saved
If the user has save system (*SAVSYS) special authority, the following authorities are not needed. When saving user profiles, *SAVSYS special authority is required.

> *Object Authority*
>> *OBJEXIST
>
> *Library Authority*
>> *EXECUTE
>
> *Object Lock*
>> *SHRNUP
>
> *Library Lock*
>> *SHRUPD

Note: Lower levels of locking may be used for objects in certain cases. See Save while active object locking rules in the Backup and Recovery topic for more information on these special cases.

≫ *Spooled Files to Be Saved*
If the user has save system (*SAVSYS) special authority, the following authorities are not needed.

> *Output Queue Authority*
>> *OBJEXIST
>
> *Output Queue Library Authority*
>> *EXECUTE
>
> *Output Queue Lock*
>> *EXCLRD

Note: Additional authority may be needed to change spooled file attributes. See "New Attributes Format" on page 188 for more information. ≪

Devices

> *Save File Authority*
>> *USE and *ADD
>
> *Save File Library Authority*
>> *EXECUTE
>
> *Save File Lock*
>> *EXCLRD
>
> *Tape or OpticalAuthority*
>> *USE
>
> *Tape or OpticalLock*
>> *EXCL
>
> *Media Library Device Lock*
>> *SHRUPD
>
> *Media Definition Authority*
>> *USE
>
> *Media Definition Library Authority*
>> *EXECUTE
>
> *Media Definition Lock*
>> *EXCLRD

> *Auxiliary Storage Pool (ASP)*
>   *USE

Note: If the save file will be cleared, *OBJMGT authority is also required.

Save While Active

> *Message Queue Authority*
>   *OBJOPR and *ADD

> *Message Queue Library Authority*
>   *EXECUTE

Output Files

> *Output File Lock*
>   *SHRRD

If the output file does not exist:

> *Output File Library Authority*
>   *READ and *ADD

If the output file exists and a new member will be added:

> *Output File Authority*
>   *OBJMGT, *OBJOPR, and *ADD

> *Output File Library Authority*
>   *EXECUTE and *ADD

If the output file exists and an existing member will be appended:

> *Output File Authority*
>   *OBJMGT and *ADD

> *Output File Library Authority*
>   *EXECUTE

If the output file exists and an existing member will be replaced:

> *Output File Authority*
>   *OBJMGT, *OBJOPR, *ADD, and *DLT

> *Output File Library Authority*
>   *EXECUTE

Required Parameter Group

Qualified user space name
    INPUT; CHAR(20)

The user space that is to hold all the information for the save operation. The first 10 characters contain the user space name. The second 10 characters contain the name of the library where the user space is located. See "User Space Format" for the format of the information in the user space.

You can use the following special values for the library name. It should be noted, however, that the library name that is actually used is not passed back to the user. Care should be taken when using these special values to avoid unexpected results.

*CURLIB   The job's current library is used to locate the user space. If no library is specified as the current library for the job, the QGPL library is used.

*LIBL   The library list is used to locate the user space.

Error code
  I/O; CHAR(*)

  The structure in which to return error information. For the format of the structure, see Error Code Parameter.

User Space Format

The following defines the format for the information in the user space. For detailed descriptions of the fields in the user space format, see "Field Descriptions."

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Number of variable length records
Note: These fields repeat for each variable length record.			
		BINARY(4)	Length of variable length record
		BINARY(4)	Key
		BINARY(4)	Length of data
		CHAR(*)	Data

If you specify a data length that is longer than the key field's defined data length, the data is truncated at the right. No error message is returned.

If you specify a data length that is shorter than the key field's defined data length, an error message is returned for binary fields. If the field is a character field, the data is padded with blanks.

Note: This does not apply to keys that allow a list of values to be specified. In these cases, the amount of data read is based on the specified number of entries in the list.

If keys are duplicated in the user space, only the last value for a given key is used for the save operation.

It is recommended, but not required, to align each variable length record on a 4-byte boundary. That is, you should make the length of each variable length record a multiple of 4, even if the data length is not a multiple of 4.

Field Descriptions

Data. The data used to specify the value for the given key.

Key. The parameter of the Save Object (SAVOBJ) command to specify. See "Valid Keys" on page 170 for the list of valid keys.

Length of data. The length of the data used to specify the value for the given parameter.

Length of variable length record. The length of the variable length record.

Number of variable length records. The number of variable length records that are passed in the user space. ≫ The valid range is from 2 through 36. ≪

Valid Keys

The following table lists the valid keys for the key field area of the variable length record. For detailed descriptions of the keys, see "Field Descriptions" on page 171.

Some messages for this API refer to parameters and values of the Save Object (SAVOBJ) command. This table can also be used to locate the key names that correspond to the SAVOBJ command parameters. The field descriptions contain, in addition to detailed descriptions, the corresponding parameter values.

The library key and the device key are required keys. The other keys are optional.

Key	Type	Field	SAVOBJ Command Parameter
1	CHAR(*)	Object information	OBJ, OBJTYPE
2	CHAR(*)	Library	LIB
3	CHAR(*)	Device	DEV
4	CHAR(20)	Save file	SAVF
5	CHAR(1)	Update history	UPDHST
6	CHAR(*)	Volume identifier	VOL
7	BINARY(4)	Sequence number	SEQNBR
8	CHAR(*)	Label	LABEL
9	CHAR(7)	Expiration date	EXPDATE
10	CHAR(1)	End of media option	ENDOPT
11	CHAR(10)	Target release	TGTRLS
12	CHAR(1)	Clear	CLEAR
13	CHAR(1)	Object precheck	PRECHK
14	CHAR(1)	Save while active	SAVACT
15	BINARY(4)	Save while active wait time for object locks	SAVACTWAIT
16	CHAR(20)	Save while active message queue	SAVACTMSGQ
17	CHAR(*)	File member	FILEMBR
18	CHAR(1)	Save access paths	ACCPTH
19	CHAR(1)	Save file data	SAVFDTA
20	CHAR(1)	Storage	STG
21	CHAR(1)	Data compression	DTACPR
22	CHAR(1)	Data compaction	COMPACT
23	CHAR(1)	Output	OUTPUT
24	CHAR(20)	Qualified output file	OUTFILE
25	CHAR(11)	Output member	OUTMBR
26	CHAR(1)	Type of output information	INFTYPE
27	CHAR(*)	Optical file	OPTFILE

Key	Type	Field	SAVOBJ Command Parameter
28	CHAR(1)	Use optimum block size	USEOPTBLK
29	CHAR(*)	Omit library	OMITLIB
30	CHAR(*)	Omit object information	OMITOBJ
31	CHAR(20)	Media definition	MEDDFN
32	CHAR(10)	ASP device name	ASPDEV
33	BINARY(4)	Save while active wait time for pending record changes	SAVACTWAIT
34	BINARY(4)	Save while active wait time for other pending changes	SAVACTWAIT
≫ 35	CHAR(*)	Spooled file data	SPLFDTA
45	CHAR(*)	Queue data	QDTA ≪

Field Descriptions

The values shown in parentheses are the corresponding values for the SAVOBJ command parameters.

ASP device name. The names of the auxiliary storage pool (ASP) devices to be included in the save operation. When saving user profiles, these are the ASPs from which private authorities are saved. The default is *. The possible values are:

*	The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the job has a linked ASP group, all independent ASPs in the linked ASP group.
*ALLAVL	The operation includes the system ASP, all basic user ASPs, and all available independent ASPs.
	Note: This value is valid only when saving user profiles.
*SYSBAS	The operation includes the system ASP and all basic user ASPs.
*CURASPGRP	If the job has a linked ASP group, all independent ASPs in the linked ASP group are included in the save operation.
ASP device name	The operation includes the specified independent ASP.

Clear. Whether active data on the media is cleared or replaced automatically. 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 the tape. Replacing active data on optical media replaces only the optical files created by this operation. The default is 0.

Notes:

1. Clearing a tape does not initialize it. Before the save command is issued, you should initialize the tape to a standard label format by using the Initialize Tape (INZTAP) command and specifying a value on the NEWVOL parameter.

2. Clearing an optical volume does initialize it.

3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

The possible values are:

0	None of the media is cleared automatically. If the save operation encounters active data on a tapeor 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. (*NONE)

1   All of the media is cleared automatically. (*ALL)

        If tapes are used and a sequence number is specified for the sequence number key, the first tape is cleared beginning at that sequence number. All tapes following the first tape are completely cleared. To clear the entire first tape, 1 must be specified for the sequence number key.
2   All media after the first volume is cleared automatically. If the save operation encounters active data on the first tape, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file. (*AFTER)

        Note: This value is not valid for save files.
3   Active data on the media is replaced automatically. Optical volumes are not initialized. Tapesand save files are cleared automatically in the same way as the value 1. (*REPLACE)

Data compaction. Whether data compaction is used. The default is 1. The possible values are:

0   Device data compaction is not performed. (*NO)
1   Device data compaction is performed if the data is saved to tape and all tape devices specified support the compaction feature. (*DEV)

Data compression. Whether data compression is used. The default is 2. The possible values are:

0   No data compression is performed. (*NO)
1   If the save operation is to tape and the target device supports compression, hardware compression is performed. If compression is not supported on the device, or if the save data is written to opticalor save file, software compression is performed. Low (SNA) software compression is used for all devices except optical DVD, which uses medium (TERSE) software compression. (*YES)
2   If the save operation is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed. (*DEV)

        Note: Note: If 2 is specified for the data compression key and 1 is specified for the data compaction key, only device data compaction is performed if compaction is supported on the device. Otherwise, data compression is performed if supported on the device.
3   If the save operation is to a save file or optical, low (SNA) 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. Low compression is usually faster than medium or high compression. The compressed data is usually larger than if medium or high compression is used. (*LOW)
4   If the save operation is to a save file or optical, medium (TERSE) 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. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used. (*MEDIUM)
5   If the save operation is to a save file or optical, high (LZ1) 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. High compression is usually slower than low and medium compression.The compressed data is usually smaller than if low or medium compression is used. (*HIGH)

Device. The names of the devices used for the save operation. The device must already be known on the system by a device description. For the format of this field, see "Device Key Format" on page 178.

End of media option. The operation that is performed automatically on the tape or optical volume after the save operation ends. If more than one volume is used, this key applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached. The default is 0.

Note: This parameter is valid only if a tape or optical device name is specified. For optical devices, 2 is the only value supported; 0 and 1 are ignored.

The possible values are:

0   The tape is automatically rewound, but not unloaded, after the operation ends. (*REWIND)
1   The tape does not rewind or unload after the operation ends. It remains at the current position on the tape
        drive. (*LEAVE)
2   The tape is automatically rewound and unloaded after the operation ends. Some optical devices eject the
        volume after the operation ends. (*UNLOAD)

Expiration date. The expiration date of the tape fileor optical file created by the save operation. If a date
is specified, the file is protected and cannot be overwritten until the specified expiration date. The default
is 0999999. The possible values are:

0999999   The file is protected permanently. (*PERM)
Expiration date   The date when protection for the file ends, specified in the format CYYMMDD:

          C   Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.

          YY   Year

          MM   Month

          DD   Day

File member. A list of the database files and their members that are to be saved. Each database file
specified here must also be specified in the list of objects to be saved. If this key is not specified, the
default of *ALL will be used for both the file name and the member name. For the format of this field,
see "File Member Format" on page 178.

Label. The name that identifies the data file on the tape. Although the label key is defined as CHAR(*),
the maximum length of a label is currently 17. If the length of data field is specified as more than 17, the
label is truncated such that only the first 17 characters are used. The default is *LIB.

*LIB   The file label is created by the system using the name of the library specified for the library key.
Data file identifier   The data file identifier of the data file used. This option is valid only for a single-library save
                     operation.

Library. A list of libraries that contain the objects that are saved. If more than one library is specified,
*ALL must be the only object name specified (object information key) and the device cannot be *SAVF.
For the format of this field, see "Library Key Format" on page 179.

Media definition. The name and library of the media definition that identifies the devices and media
used to contain the saved data. For information about creating and using a media definition, see Save to
multiple devices in the Backup and recovery topic and the "Create Media Definition (QSRCRTMD,
QsrCreateMediaDefinition) API" on page 14 (QSRCRTMD, QsrCreateMediaDefinition) API. The first 10
characters contain the media definition name; the second 10 characters contain the library in which the
media definition is located.

You can use these special values for the library name:

*CURLIB   The job's current library is used to locate the media definition. If no library is specified as the
                     current library for the job, the QGPL library is used.
*LIBL   The library list.

Object information. A list of the name and type of the objects to be saved. If *ALL is specified for the
object name and object type, the list cannot contain other entries. The default for both the object name
and the object type is *ALL. For the format of this field, see "Object Information Format" on page 180.

Object precheck. Whether the save operation for a library should end if all objects specified by the API do not satisfy all 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

The default is 0. The possible values are:

0	The save operation for a library continues, saving only those objects that can be saved. (*NO)
1	If one or more objects cannot be saved after the specified objects are checked, the save operation for a library ends before any data is written. (*YES)

Omit libraries. A list of the libraries to be omitted from the save operation. The default is *NONE. For the format of this field, see "Omit Library Key Format" on page 180.

Omit object information. A list of the name and type of the objects and library to be omitted from the save operation. If *ALL is specified for the object name and object type, the list cannot contain other entries. The default for both the object name and the object type is *ALL. For the format of this field, see "Omit Object Information Format" on page 181.

Optical file. The name that identifies the file on the optical volume. Although the optical file is defined as CHAR(*), the maximum length of an optical file name is currently 256 characters. If the length of data field is specified as more than 256 characters, the name is truncated such that only the first 256 characters are used. The default is '*'. The possible values are:

''*	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	The path name of the optical file that is used for the save operation, beginning with the root directory of the volume.

Output. Whether a list of information about the saved objects is created. The default is 0. The possible values are:

0	No output listing is created. (*NONE)
1	The output is printed with the job's spooled output. (*PRINT)
2	The output is directed to the database file specified with the output file key. (*OUTFILE)

Output member. The name of the database file member used to save the object information. This field also determines whether to replace or add the data if the member already exists. The defaults are *FIRST for the output member name field and 0 for the option field. For the format of this field, see "Output Member Format" on page 181.

Qualified output file. The qualified name of the database file to which the information about the objects is directed. This key is required only if the output key is set to 2. The first 10 characters contain the output file name; the second 10 characters contain the output file library. The possible values for output file library are:

CURLIB	The job's current library is used to locate the output file. If no library is specified as the current library for the job, the QGPL library is used.
LIBL	The library list is used to locate the output file.
Library name	The name of the library where the output file is located.

≫ **Queue data.** A description of queue data to be saved. The default is no queue data. For the format of this key, see Queue Data Key Format (page "Queue Data Key Format" on page 182). ≪

Save access paths. Whether the logical file access paths that are dependent on the physical files being saved are also saved. The default is 2. The possible values are:

0   The logical file access paths are not saved. (*NO)
1   The specified physical files and all eligible logical file access paths over them are saved. (*YES)
2   The system value QSAVACCPTH determines whether to save the logical file access paths that are dependent on the physical files that are being saved. (*SYSVAL)

Save file. The name and library of the save file that is used to contain the saved data. The first 10 characters contain the save file name; the second 10 characters contain the library where the save file is located.

You can use these special values for the library name:

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

Save file data. For save file objects, whether only the description of a save file, or both the description and the contents of a save file are saved. The default is 1. The possible values are:

0   Only the description of a save file is saved. (*NO)
1   The description and contents of a save file are saved. (*YES)

Note: For System/38 environments, the default value of 1 is not valid; therefore, this key must explicitly be set to a value of 0.

Save while active. Whether an object can be updated while it is being saved. The default is 0. The possible values are:

0   Objects that are in use are not saved. (*NO)
1   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. (*SYSDFN)
2   Objects in a library can be saved while they are in use by another job. All the objects in a library reach a checkpoint together. They are saved in a consistent state in relationship to each other. (*LIB)
3   Objects in a library can be saved while they are in use by another job. All the objects and all the libraries in the save operation reach a checkpoint together. They are saved in a consistent state in relationship to each other. (*SYNCLIB)

Save while active message queue. The name and library of the message queue that is used to notify the user that the checkpoint processing for a library is complete. The first 10 characters contain the message queue name; the second 10 characters contain the name of the library where the message queue is located. If *NONE or *WRKSTN is specified for the message queue name, blanks must be specified for the message queue library. The defaults are *NONE for the message queue name and blanks for the library.

The possible values for the message queue name are:

*NONE   No notification message is sent.
*WRKSTN   The notification message is sent to the work station message queue. This is not valid in batch mode.

Message queue name   The name of the message queue.

The possible values for the message queue library are:

*CURLIB	The job's current library is used to locate the message queue. If no library is specified as the current library for the job, the QGPL library is used.
*LIBL	The library list is used to locate the message queue.
Library name	The name of the library where the message queue is located.

Save while active wait time for object locks. If an object is not available, the amount of time to wait for a lock on the object before continuing the save operation. The default is 120. The possible values are:

-1	No maximum wait time exists.
0-99999	The time (in seconds) to wait.

Save while active wait time for other pending changes. For each library, the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.
- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API.

If a commit boundary is not reached for a library in the specified time, library is not saved. The default is -2. The possible values are:

-1	No maximum wait time exists. (*NOMAX)
-2	The system waits up to the value specified on the Save while active wait time for object locks key for the types of transactions that are listed above to reach a commit boundary. (*LOCKWAIT)
0-99999	The time (in seconds) to wait. If 0 is specified, and only one object is specified on the Object information key, and that object is a file, then the system will save the object without requiring the types of transactions that are listed above to reach a commit boundary.

Save while active wait time for pending record changes. For each group of objects that are checkpointed together, the amount of time to wait for transactions with pending record changes to reach a commit boundary. The Save while active key determines which objects are checkpointed together. If 0 is specified, all objects being saved must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value -3 is specified. The default is -2. The possible values are:

-1	No maximum wait time exists. (*NOMAX)
-2	The system waits up to the value specified on the Save while active wait time for object locks key for transactions with pending record changes to reach a commit boundary. (*LOCKWAIT)
-3	The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions. (*NOCMTBDY)
	If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify this value.
	Note: This value cannot be specified if the Target release key is earlier than V5R3M0.

0-99999   The time (in seconds) to wait.

Sequence number. The sequence number to use for the save operation when tape is used. The default is -1. The possible values are:

-1   The save operation begins after the last sequence number on the tape volume.
1-16777215   The sequence number of the file to be used for the save operation.

≫ **Spooled file data.** A description of spooled file data to be saved. The default is no spooled file data. For the format of this key, see "Spooled File Data Key Format" on page 182. ≪

Storage. Whether the system storage that is occupied by the data portion of the following objects in the library being saved is freed:
- Files
- Modules
- Programs
- Service programs
- Structured Query Language (SQL) packages
- Journal receivers

The default is 0. The possible values are:

0   The storage occupied by the data portion of the objects is not freed. (*KEEP)
1   The storage occupied by the data portion of the objects is freed. The storage is freed only after all the objects in the library are saved successfully. (*FREE)

Target release. The release of the operating system on which the objects will be restored and used. The object types specified (in the object information field) must exist on the specified release. The default is *CURRENT. The possible values are:

*CURRENT   The objects are restored to, and used on, the release of the operating system currently running on the system.
*PRV   The objects are to be restored on the previous release that has modification level 0 of the operating system.
Release level   The release level in the format VxRxMx, where x is the number of the version, release, and modification level.

Type of output information. The type of information that is printed or directed to the output database file. The default is 0. The possible values are:

0   The list contains an entry for each object requested to be saved. (*OBJ)
1   The list contains an entry for each library requested to be saved. (*LIB)
2   The list contains an entry for each object, database file member, ≫ and spooled file ≪ requested to be saved. (*MBR)
3   The list contains an entry for each library that is requested to be saved and an entry for each object that was not successfully saved. (*ERR)

Update history. Whether the save history information of each object is changed to the date, time, and location of this save operation. The default is 1. The possible values are:

0   The save history information of each object saved is not updated. (*NO)

1   The last save date, time, and location is updated in each object saved. (*YES)

Use optimum block size. Whether the tape device's optimum block size should be used. The default is 0. The possible values are:

0   The tape device's optimum block size is not used. A block size that is compatible with all i5/OS releases and tape devices is used. (*NO)

1   The tape device's optimum block size is used. The system may create a tape that is only compatible with a tape device that supports the same block size. Performance will likely but not necessarily improve. Commands such as Duplicate Tape (DUPTAP) do not duplicate the tape unless it is being duplicated to a tape device that supports the same block size. (*YES) Data compression is ignored when optimum block size is used.

Volume identifier. The volume identifiers of the tape volumesor optical volumes on which the object data is to be saved. All volume identifiers must be entered in the order in which you want them saved. The default is *MOUNTED. For the format of this field, see "Volume Identifier Format" on page 189.

Device Key Format

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Number in array
Note: This field repeats for each device name.			
		CHAR(10)	Device name

Field Descriptions

Device name. The name of the device used for the save operation. The possible values for each element of the array are:

*SAVF   The save operation is done using the save file specified by the save file key. If specified, it must be the only element in the array.

*MEDDFN   The save operation is done by using the devices and media that are identified in the media definition, which is specified by the media definition key. If specified, it must be the only element in the array.

Media library   The name of the media library device used for the save operation. If specified, it must be the only
device name   element in the array.

Optical device   The name of the optical device used for the save operation. If specified, it must be the only
name   element in the array.

Tape device name   The name of the tape device used for the save operation. A maximum of four tape devices may be used. They must be specified in the order in which they should be used.

Number in array. The number of devices to be used during the save operation. The possible values are 1 through 4.

File Member Format

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Number in array

Offset			
Dec	Hex	Type	Field
Note: These fields repeat for each file.			
		CHAR(10)	File name
		CHAR(2)	Reserved
		BINARY(4)	Number of members
Note: This field repeats for each member associated with the given file.			
		CHAR(10)	Member

Field Descriptions

File name. The name of the file being saved. The possible values are:

*ALL	The list of member names that follow this value applies to all files indicated in the list of objects to save. If *ALL is specified for the file name, it must be the only file name in the list.
Database file name	The name of the database file from which the listed members are saved.

Member. The name of the member to save. The possible values are:

*ALL	All members are saved from the specified file. If *ALL is specified for member name, it must be the only member name for that file.
*NONE	No members are saved from the specified file. Only the file description is saved.
Member name	The name of the member to save. It may be either a simple name or a generic name.

Number in array. The number of file and member structures used during the save operation. The possible values are 1 through 50.

Number of members. The number of member names for the given file name. Possible values are 1 through 50.

Reserved. An ignored field.

Library Key Format

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
Note: This field repeats for each library name.			
		CHAR(10)	Library name

Field Descriptions

Library name. The name of the library containing the objects. The possible values are:

≫ *SPLF	Spooled file data is to be saved. If this value is specified, it must be the only element in the array, the spooled file data key must be specified, and *ALL must be specified for the object name and object type. ≪
Library name	Either a simple or generic library name

Number in array. The number of libraries used during the save operation. ≫ The possible values are 1 through 32767. ≪

Object Information Format

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Number in array
Note: These fields repeat for each object name.			
		CHAR(10)	Object name
		CHAR(10)	Object type

Field Descriptions

Number in array. The number of objects that are specified for this key. There is no limit for the number in array field. The total amount of information in the user space, however, cannot exceed 16MB.

Object name. The name of the object that is to be saved. The possible values are:

*ALL	All the objects in the specified libraries, depending on the values specified for object type
Object name	Either a simple name or a generic name

Object type. The type of the object that is to be saved. The possible values are:

*ALL	All objects with the specified object name that are valid types for the SAVOBJ command on the current release of the system.
*USRPRF	All user profiles with the specified object name. If this value is specified, all entries in the object information key must specify this type and the saved data must be restored with the RSTUSRPRF command.
Object type	A valid type for the SAVOBJ command on the current release of the system

Omit Library Key Format

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Number in array
Note: This field repeats for each library name.			
		CHAR(10)	Library name

Field Descriptions

Library name. The name of the library containing the objects to omit. The possible values are:

*NONE	No libraries are excluded from the save operation.
Library name	Either a simple or generic library name

Number in array. The number of libraries to omit from the save operation. ≫ The possible values are 1 through 32767. ≪

Omit Object Information Format

Offset			
Dec	**Hex**	**Type**	**Field**
		BINARY(4)	Number in array
Note: These fields repeat for each object name.			
		CHAR(10)	Object name
		CHAR(10)	Library name
		CHAR(10)	Object type

Field Descriptions

Library name. The name of the library that is to be omitted. The possible values are:

*ALL   All the libraries, depending on the values specified for object and object type
Library name   Either a simple name or a generic name

Number in array. The number of values that are specified for this key. » The possible values are 1 through 32767. «

Object name. The name of the object that is to be omitted. The possible values are:

*ALL   All the objects in the specified libraries, depending on the values specified for object type
Object name   Either a simple name or a generic name

Object type. The type of the object that is to be omitted. The possible values are:

*ALL   All objects with the specified object name that are valid types for the SAVOBJ command on the current release of the system
*USRPRF   All user profiles with the specified object name.
Object type   A valid type for the SAVOBJ command on the current release of the system

Output Member Format

Offset			
Dec	**Hex**	**Type**	**Field**
		CHAR(10)	Output member name
		CHAR(1)	Option

Field Descriptions

Option. An indicator of whether to add to or replace the existing member. The possible values are:

0   The existing records in the specified database file member are replaced by the new records. (*REPLACE)
1   The new records are added to the existing information in the database file member. (*ADD)

Output member name. The name of the file member that receives the output. The possible values are:

FIRST   The first member in the file is used and receives the output.
Member name   If the member does not exist, the system creates it.

Queue Data Key Format

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Number in array
Note: This field repeats for each queue data value.			
		BIN(4)	Queue data

Field Descriptions

Queue data. For queue objects, whether only the description of a queue, or both the description and the contents of a queue are saved. The default is 0.

The possible values are:

0   Only the description of a queue is saved. (*NONE)
1   The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved. (*DTAQ)

Number in array. The number of queue data values. The possible value is 1.

Spooled File Data Key Format

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Spooled file data
4	4	BINARY(4)	Length of spooled file data header
8	8	BINARY(4)	Offset to spooled file selection list

Field Descriptions

Length of spooled file data header. The length of the spooled file data header information. The possible values are:

8   The header information ends with the length field.
12   The header information ends with the offset to selection list field.

Offset to spooled file selection list. The offset from the start of the user space to the first "Spooled File Selection List Entry Format" on page 183. The default is 0. If the value of the spooled file data field is 2, the value of this field must be greater than 0. Otherwise, the value must be 0.

Spooled file data. Whether to save spooled file data and attributes. The default is 0. The possible values are:

0	No spooled file data is saved. (*NONE)
1	For each output queue that is saved, all available spooled file data on the output queue is saved. (*ALL)
2	Selected spooled file data is saved. The offset to selection list field must be specified.

Spooled File Selection List Entry Format

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Length of spooled file selection list entry
4	4	BINARY(4)	Offset to next spooled file selection list entry
8	8	BINARY(4)	Include or omit
12	C	BINARY(4)	Selection criteria format
16	10	BINARY(4)	Offset to selection criteria
20	14	BINARY(4)	Offset to new attributes

Field Descriptions

Include or omit. Whether the spooled files selected by this entry are included or omitted from the save operation. Omit takes precedence over include. The possible values are:

0	Spooled files that match all of the values specified in the selection criteria are omitted from the save operation.
1	Spooled files that match all of the values specified in the selection criteria are included in the save operation, unless another entry omits them. At least one entry must have this value.

Length of spooled file selection list entry. The length of the spooled file selection list entry information. The possible values are:

20	The selection list entry ends with the offset to selection criteria field.
24	The selection list entry ends with the offset to new attributes field.

Offset to new attributes. The offset from the start of the user space to the new attributes for the spooled files included by this selection list entry. The value must be 0 if the Include or omit field value is 0. For the format of the new attributes, see "New Attributes Format" on page 188.

Offset to next spooled file selection list entry. The offset from the start of the user space to the next spooled file selection list entry. The value must be 0 for the last entry in the list.

Offset to selection criteria. The offset from the start of the user space to the selection criteria.

Selection criteria format. The format of the spooled file selection criteria. The possible values are:

1	The selection criteria is specified by the "Spooled File ID Format" on page 184. This format identifies exactly one spooled file.
2	The selection criteria is specified by the "Spooled File Attributes Format" on page 185. This format identifies any number of spooled files.

Spooled File ID Format

This is the format of the spooled file selection criteria when a value of 1 is specified for the selection criteria format field. The criteria specified must uniquely identify a single spooled file.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Length of Spooled file ID
4	4	CHAR(26)	Qualified job name
30	1E	CHAR(10)	Spooled file name
40	28	BINARY(4)	Spooled file number
44	2C	CHAR(8)	Job system name
52	34	CHAR(7)	Creation date
59	3B	CHAR(6)	Creation time

Field Descriptions

Creation date. The date the spooled file was created. This value is considered after the qualified job name, spooled file name, spooled file number, and job system name values. The possible values are:

*LAST   The spooled file with the latest creation date and time for the specified qualified job name, spooled file name, spooled file number, and job system name is selected.

*ONLY   There is only one spooled file with the specified qualified job name, spooled file name, spooled file number, and job system name.

Date   The date the spooled file was created, in the format CYYMMDD:

    C   Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.

    YY Year

    MM Month

    DD Day

Creation time. The time the spooled file was created. This field must be set to blanks if *LAST or *ONLY is specified for creation date. This value is considered after the qualified job name, spooled file name, spooled file number, job system name, and creation date values. The possible values are:

*LAST   The spooled file with the latest creation time for the specified qualified job name, spooled file name, spooled file number, job system name, and creation date is selected.

*ONLY   There is only one spooled file with the specified qualified job name, spooled file name, spooled file number, job system name, and creation date.

Time   The time the spooled file was created, in the format HHMMSS:

    HH Hour

    MM Minute

    SS Second

Job system name. The name of the system where the job that owns the spooled file ran. This value is considered after the qualified job name, spooled file name, and spooled file number values. The possible values are:

*ONLY	There is only one spooled file with the specified qualified job name, spooled file name, spooled file number, creation date, and creation time.
*CURRENT	The spooled file created on the current system with the specified qualified job name, spooled file name, spooled file number, creation date, and creation time is selected.
*ANY	The job system name is not considered when selecting a spooled file. Use this value when you want the creation date and creation time values to take precedence over the job system name when selecting a spooled file.
System name	The name of the system where the job that owns the spooled file ran.

Length of spooled file ID. The length of the spooled file ID information. The possible values are:

65	The spooled file ID ends with the creation time field.

Qualified job name. The name of the job that owns the spooled file. The qualified job name has three parts:

Job name	CHAR(10)	A specific job name or the following special value: * Current job. The rest of the qualified job name must be blank.
User name	CHAR(10)	A specific user profile name or blanks when the job name is *.
Job number	CHAR(6)	A specific job number or blanks when the job name is *.

Spooled file name. The name of the spooled file.

Spooled file number. The unique number of the spooled file. The possible values are:

0	There is only one spooled file with the specified qualified job name and spooled file name. (*ONLY)
-1	The spooled file with the highest number for the specified qualified job name and spooled file name is selected. (*LAST)
-2	The spooled file number is not considered when selecting a spooled file. Use this value when you want the system name value or spooled file create date and spooled file create time values to take precedence over the spooled file number when selecting a spooled file. (*ANY)
1-999999	The number of the spooled file for the specified qualified job name and spooled file name.

Spooled File Attributes Format

This is the format of the spooled file selection criteria when a value of 2 is specified for the selection criteria format field.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Length of spooled file attributes
4	4	CHAR(20)	Qualified output queue
24	18	CHAR(10)	Spooled file name
34	22	CHAR(10)	Job name
44	2C	CHAR(10)	User name
54	36	CHAR(6)	Job number
60	3C	CHAR(10)	User-specified data

Offset		Type	Field
Dec	Hex		
70	46	CHAR(8)	Job system name
78	4E	CHAR(10)	Form type
88	58	CHAR(13)	Starting creation date and time
101	65	CHAR(13)	Ending creation date and time

Field Descriptions

Ending creation date and time. Spooled files with a creation date and time less than or equal to this date and time are selected. The default is *ALL. The following special value is allowed:

*ALL   Ending creation date and time are not used to select spooled files.

The date and time must be specified in the format CYYMMDDHHMMSS:

C   Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.

YY   Year

MM   Month

DD   Day

HH   Hour

MM   Minute

SS   Second

Form type. Spooled files with this form type are selected. Either a specific value or generic value may be specified. The default is *ALL. The following special values are allowed:

*ALL   Spooled files with any form type are selected.
*STD   Spooled files that specify the standard form type are selected.

Job name. Spooled files owned by this job are selected. Either a specific name or generic name may be specified. The default is *ALL. The following special value is allowed:

*ALL   Spooled files owned by any job are selected.

Job number. Spooled files owned by a job with this job number are selected. If a job number is specified, then a specific job name and a specific user name must also be specified. The default is *ALL. The following special value is allowed:

*ALL   Spooled files owned by a job with any job number are selected.

Job system name. Spooled files owned by a job on this system are selected. Either a specific name or generic name may be specified. The default is *ALL. The following special values are allowed:

*ALL   Spooled files created on any system are selected.
*CURRENT   Spooled files owned by a job on the current system are selected.

Length of spooled file attributes. The length of the spooled file data attributes information. The possible values are:

24	The spooled file attributes end with the qualified output queue field.
34	The spooled file attributes end with the spooled file name field.
44	The spooled file attributes end with the job name field.
54	The spooled file attributes end with the user name field.
60	The spooled file attributes end with the job number field.
70	The spooled file attributes end with the user-specified data field.
78	The spooled file attributes end with the job system name field.
88	The spooled file attributes end with the form type field.
101	The spooled file attributes end with the starting creation date and time field.
114	The spooled file attributes end with the ending creation date and time field.

Qualified output queue. Spooled files on this output queue are selected, if they are found in the ASPs specified for the ASP device key. The qualified output queue has two parts:

Object name   CHAR(10). A specific or generic output queue name or the following special value:

> **ALL*   Spooled files on all output queues that satisfy the library name are selected.

Library name   CHAR(10). A specific or generic library name, or one of the following special values:

> **ALL*   All libraries.
>
> **CURLIB*
> The job's current library. If no library is specified as the current library for the job, the QGPL library is used.
>
> **LIBL*   The libraries in the library list.

Spooled file name. Spooled files with this name are selected. Either a specific name or generic name may be specified. The default is *ALL. The following special value is allowed:

*ALL   Spooled files with any name are selected.

Starting creation date and time. Spooled files with a creation date and time greater than or equal to this date and time are selected. The default is *ALL. The following special value is allowed:

*ALL   Starting creation date and time are not used to select spooled files.

The date and time must be specified in the format CYYMMDDHHMMSS:

C   Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.

YY   Year

MM   Month

DD   Day

HH   Hour

MM   Minute

SS   Second

User name. Spooled files owned by this user are selected. Either a specific name or generic name may be specified. The default is *ALL. The following special value is allowed:

*ALL   Spooled files with any user are selected.

User-specified data. Spooled files with this user-specified data value are selected. Either a specific value or generic value may be specified. The default is *ALL. The following special value is allowed:

*ALL   Spooled files with any user-specified data value are selected.

New Attributes Format

This is the format of new attributes to be assigned to the selected spooled files.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Length of new attributes
4	4	BINARY(4)	Expiration days

Field Descriptions

Expiration days. The number of days from the start of the operation when the selected spooled files will expire. The expiration date will be set for the spooled files on the system after they have been successfully saved. The default is 0.

Note: The user needs additional authority to use any value other than 0. The default value of 0 will be used for any spooled files which the user is not authorized to change. The user is authorized to change the expiration date of a spooled file if any of the following conditions are met.

- The user owns the spooled file.
- The user has spool control (*SPLCTL) special authority.
- The user has job control (*JOBCTL) special authority, and the output queue on which the spooled file resides is specified as OPRCTL(*YES).
- The user owns the output queue on which the spooled file resides, and the output queue is specified as AUTCHK(*OWNER).
- The user has read, add, and delete authorities to the output queue on which the spooled file resides, and the output queue is specified as AUTCHK(*DTAAUT).

The possible values are:

-1   The expiration date for the selected spooled files will be set to *NONE (no expiration date).
0   The expiration date for the selected spooled files will not be changed.
1-366   The expiration date for the selected spooled files will set to the number of days specified past the date that the save operation begins.

Length of new attributes. The length of the new attributes information. The possible values are:

8   The new attributes end with the expiration days field.

≪

Volume Identifier Format

Offset			
Dec	Hex	Type	Field
		BINARY(4)	Number in array
Note: These fields repeat for each volume identifier.			
		BINARY(4)	Length of volume identifier
		CHAR(*)	Volume identifier

Field Descriptions

Length of volume identifier. The character length of the identifier of the volume. The possible value is:

n	The size of a single volume identifier. The maximum size of a tapevolume identifier is 6 characters. The maximum size of an optical volume identifier is 32 characters. If a volume identifier larger than the maximum size is entered for this key, it is truncated to the maximum size. If the volume identifier is *MOUNTED, this value must be 8.

Number in array. The number of volume identifiers used during the save operation. The possible values are 1 through 75.

Volume identifier. The identifier of a volume. The possible values are:

*MOUNTED	The volume currently placed in the device is used. If *MOUNTED is specified, it must be the only value specified. This value cannot be specified for an optical media library device. *MOUNTED cannot be specified for a tape media library device unless a category is set with the Set Tape Category (SETTAPCGY) command.
Volume identifier	The identifier of a volume.

Dependencies between Keys

The following two tables list the dependencies between the different keys. If the dependency holds only for a certain value, then that value is also shown (key = *n*, where *n* is the value). Otherwise, if the dependency is true for all values of the key, then only the name of the key is given.

The following table lists the conditions where specifying a certain key forces the use of another key.

If you specify...	...must be specified
Multiple library names or generic library	Object name = *ALL
Device = tape device	Volume identifier [1] Sequence number [1] Label [1] Expiration date [1] End of media option [1]
Device = optical device	Volume identifier Optical file [1] Expiration date [1]
Device = media definition	Media definition
Output = 1	Type of output information [1]

If you specify...	...must be specified
Output = 2	Output file Output member [1] Type of output information [1]
Save while active = 1, 2, or 3	Save while active wait time for object locks [1] Save while active wait time for pending record changes [1] Save while active wait time for other pending changes [1] Save while active message queue [1]
Storage = 1	Save while active = 0 [1]
Object type = *USRPRF	Library = QSYS Label = QFILEUPR[1]
» Library name = *SPLF	Object name = *ALL[1] Object type = *ALL[1] Spooled file data = 2
Library name <> *SPLF	Spooled file data = 0[1] or 1 «
Notes:	
1. This key does not have to be explicitly specified. The default may be taken to satisfy this dependency.	

The following table lists the conditions where specifying a certain key excludes the user from using another key, or a particular value of that key.

If you specify...	...cannot be specified
Save file	Volume identifier Sequence number Label Expiration date End of media option Clear = 2 Optical file Use optimum block size Media definition
Media definition	Volume identifier Sequence number Optical file
Tape, optical, or media definition for the device	Save file
Save while active = 0	Save while active wait time for object locks Save while active wait time for pending record changes Save while active wait time for other pending changes Save while active message queue
Output = 0	Output file Output member Type of output information
Optical file	Label Use optimum block size

If you specify...	...cannot be specified
Object type = *USRPRF	Any other object type Save while active Save while active wait time for object locks Save while active wait time for pending record changes Save while active wait time for other pending changes Save while active message queue File member » Queue data « Save access paths Save file data » Spooled file data « Storage Omit library Media definition
Save while active wait time for pending record changes = -3	Target release = release earlier than Version 5 Release 3
» Spooled file data <> 0	Target release = release earlier than Version 5 Release 4
Queue data = 1	Target release = release earlier than Version 5 Release 4 «

Relationship to SAVOBJ and SAVSECDTA Commands

Because of the relationship between the QSRSAVO API and the SAVOBJ and SAVSECDTA commands, the following situations should be noted:

- Message text: Several messages produced by this API refer to parameters or values of the SAVOBJ command (for example, *AFTER). To determine which key a given parameter corresponds to, see "Valid Keys" on page 170. To determine which key value a given parameter value corresponds to, see "Field Descriptions" on page 171.
- Command type: The command type listed for the API on headings of displays and print files is SAVOBJ or SAVSECDTA, not QSRSAVO.
- This API can be used to save one or more user profiles. It does not save other objects that are saved by the SAVSECDTA command, such as authorization lists and authority holders. The saved user profiles must be restored with the RSTUSRPRF command.
- This API can be used to save user profiles for a previous release; the SAVSECDTA command saves user profiles for the current release only. When a user profile is saved for a previous release, the interactive profile section of the user profile which contains session settings and product level information, is not saved.

Error Messages

Message ID	Error Message Text
CPF222E E	&1 special authority is required.
CPF24B4 E	Severe error while addressing parameter list.
CPF3700 E	All CPF37xx messages could be signalled. xx is from 01 to FF.
CPF3800 E	All CPF38xx messages could be signalled. xx is from 01 to FF.
CPF3C31 E	Object type &1 is not valid.
CPF3C4D E	Length &1 for key &2 not valid.

Message ID	Error Message Text
CPF3C81 E	Value for key &1 not valid.
CPF3C82 E	Key &1 not valid for API &2.
CPF3C83 E	Key &1 not allowed with value specified for key &2.
CPF3C84 E	Key &1 required with value specified for key &2.
CPF3C85 E	Value for key &1 not allowed with value for key &2.
CPF3C86 E	Required key &1 not specified.
CPF3C87 E	Key &1 allows one value with special value.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF5729 E	Not able to allocate object &1.
CPF9800 E	All CPF98xx messages could be signaled. xx is from 01 to FF.
CPFB8ED E	Device description &1 not correct for operation.

API introduced: V3R1

Qp0lSaveStgFree()—Save Storage Free

Syntax

```
#include <Qp0lstdi.h>

int Qp0lSaveStgFree(
  Qlg_Path_Name_T          *Path_Name,
  Qp0l_StgFree_Function_t  *UserFunction_ptr,
  void                     *Function_CtlBlk_ptr);
```

Service Program Name: QP0LLIB3
Default Public Authority: *USE
Threadsafe: Conditional; see "Usage Notes" on page 195.

The **Qp0lSaveStgFree()** function calls a user-supplied exit program to save i5/OS objects of type *STMF and, upon successful completion of the exit program, frees the storage for the object and marks the object as storage freed. The *STMF object and its attributes remain on the system, but the storage occupied by the *STMF object's data is deleted. The *STMF object cannot be used until it is restored to the system. This is accomplished by either of the following:

- Restoring the object using the RST command.
- Requesting an operation on the object, requiring one of the following, which will dynamically retrieve (restore) the *STMF object:
  - Accessing the object's data (**open()**, **creat()**, MOV, CPY, CPYFRMSTMF, or CPYTOSTMF).
  - Adding a new name to the object (RNM, ADDLNK, **link()**, **rename()**, **Qp0lRenameKeep()**, or **Qp0lRenameUnlink()**).
  - Checking out the object (CHKOUT).

The restore operation is done by calling a user-provided exit program registered against the Storage Extension exit point QIBM_QTA_STOR_EX400. For information on this exit point, see the "Storage Extension Exit Program" on page 211.

Qp0lSaveStgFree() returns EOFFLINE for an object that is already storage freed or returns EBUSY for an object that is checked out.

The user exit program can be either a procedure or a program.

Parameters

Path_Name

> (Input) A pointer to a path name whose last component is the object that is saved and whose storage is freed. This path name is in the Qlg_Path_Name_T format. For more information on this structure, see Path name format.
>
> If the last component of the path name supplied on the call to **Qp0lSaveStgFree()** is a symbolic link, then **Qp0lSaveStgFree()** resolves and follows the link to its target and performs its normal **Qp0lSaveStgFree()** functions on that target. If the symbolic link refers to an object in a remote file system, **Qp0lSaveStgFree()** returns ENOTSUP to the calling program.

UserFunction_ptr

> (Input) A pointer to a structure that contains information about the user exit program that the caller wants **Qp0lSaveStgFree()** to call to save an *STMF object. This user exit program can be either a procedure or a program. If this pointer is NULL, **Qp0lSaveStgFree()** does not call an exit program to save the object but does free the object's storage and marks it as storage freed.

User Function Pointer			
Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Function type flag
14	E	CHAR(10)	Program library
4	4	CHAR(10)	Program name
24	18	CHAR(1)	Multithreaded job action
25	19	CHAR(7)	Reserved
32	20	PP(*)	Procedure pointer to exit procedure

> **Function type flag.** A flag that indicates whether the Save Storage Free exit program called by **Qp0lSaveStgFree()** is a procedure or a program. If the exit program is a procedure, this flag is set to 0, and the procedure pointer to exit procedure field points to the procedure called by **Qp0lSaveStgFree()**. If the exit program is a program, this flag is set to 1 and a program name and program library are provided, respectively, in the program name and program library fields. Valid values follow:

0	QP0L_USER_FUNCTION_PTR: A user procedure is called.
1	QP0L_USER_FUNCTION_PGM: A user program is called.

> **Multithreaded job action.** (Input) A CHAR(1) value that indicates the action to take in a multithreaded job. The default value is QP0L_MLTTHDACN_SYSVAL. For release compatibility and for processing this parameter against the QMLTTHDACN system value, x'00, x'01', x'02', & x'03' are treated as x'F0', x'F1', x'F2', and x'F3'.

x'00'	QP0L_MLTTHDACN_SYSVAL: The API evaluates the QMLTTHDACN system value to determine the action to take in a multithreaded job. Valid QMLTTHDACN system values follow:

> '1' Call the exit program. Do not send an informational message.
>
> '2' Call the exit program and send informational message CPI3C80.
>
> '3' The exit program is not called when the API determines that it is running in a multithreaded job. ENOTSAFE is returned.

x'01'	QP0L_MLTTHDACN_NOMSG: Call the exit program. Do not send an informational message.
x'02'	QP0L_MLTTHDACN_MSG: Call the exit program and send informational message CPI3C80.

x'03'   QP0L_MLTTHDACN_NO: The exit program is not called when the API determines that it is running in a multithreaded job. ENOTSAFE is returned.

Procedure pointer to exit procedure.   If the function type flag is 0, which indicates that a procedure is called instead of a program, this field contains a procedure pointer to the procedure that **Qp0lSaveStgFree()** calls. This field must be NULL if the function type flag is 1.

Program library.   If the function type flag is 1, indicating a program is called, this field contains the library in which the program being called (identified by the program name field) is located. This field must be blank if the function type flag is 0.

Program name.   If the function type flag is 1, indicating a program is called, this field contains the name of the program that is called. The program should be located in the library identified by the program library field. This field must be blank if the function type flag is 0.

Reserved.   A reserved field. This field must be set to binary zero.

Function_CtlBlk_ptr
(Input) A pointer to any data that the caller of **Qp0lSaveStgFree()** wants to have passed to the user-defined Save Storage Free exit program that **Qp0lSaveStgFree()** calls to save an *STMF object. **Qp0lSaveStgFree()** does not process the data that is referred to by this pointer. The API passes this pointer as a parameter to the user-defined Save Storage Free exit program that was specified on its call. This is a means for the caller of **Qp0lSaveStgFree()** to pass information to and from the Save Storage Free exit program.

Authorities

The following table shows the authorization required for the Qp0lSaveStgFree() API.

Object Referred to	Authority Required	errno
Each directory, preceding the last component, in a *path name*	*RX	EACCES
Object	*SAVSYS or *RW	EACCES
Any called program pointed to by the *UserFunction_ptr* parameter	*X	EACCES
Any library containing the called program pointed to by the *UserFunction_ptr* parameter	*X	EACCES

Return Value

0   **Qp0lSaveStgFree()** was successful.

-1   **Qp0lSaveStgFree()** was not successful. The *errno* global variable is set to indicate the error.

Error Conditions

If **Qp0lSaveStgFree()** is not successful, *errno* indicates one of the following errors:

Error condition   **Additional information**

[EACCES]   If you are accessing a remote file through the Network File System, update operations to file permissions at the server are not reflected at the client until updates to data that is stored locally by the Network File System take place. (Several options on the Add Mounted File System (ADDMFS) command determine the time between refresh operations of local data.) Access to a remote file may also fail due to different mappings of user IDs (UID) or group IDs (GID) on the local and remote systems.

[EAGAIN]

[EBADNAME]

Error condition	Additional information
[EBUSY]	
[EDAMAGE]	
[EFAULT]	
[EINVAL]	
[EIO]	
[EISDIR]	
[ELOOP]	
[EMFILE]	
[ENAMETOOLONG]	
[ENFILE]	
[ENOENT]	
[ENOMEM]	
[ENOTAVAIL]	
[ENOTDIR]	
[ENOSPC]	
[ENOSYSRSC]	
[ENOTSAFE]	
[ENOTSUP]	
[EOFFLINE]	
[EUNKNOWN]	

Error Messages

The following messages may be sent from this function:

Message ID	Error Message Text
CPI3C80 I	An exit program has been called for which the threadsafety status was not known.
CPFA0D4 E	File system error occurred.
CPE3418 E	Possible APAR condition or hardware failure.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

Usage Notes

- This function will fail with error code [ENOTSAFE] when both of the following conditions occur:

  - Where multiple threads exist in the job.
  - The object this function is operating on resides in a file system that is not threadsafe. Only the following file systems are threadsafe for this function:

    - ″Root″ (/)
    - QOpenSys
    - User-defined
    - QNTC
    - QSYS.LIB
    - QOPT
    - Network File System
    - QFileSvr.400

- If the Save Storage Free exit program calls the SAV command or the **QsrSave** function or any other function that is not threadsafe, and there are secondary threads active in the job, **Qp0lSaveStgFree()** may fail as a result.
- If the Save Storage Free exit program is not threadsafe or uses a function that is not threadsafe, then **Qp0lSaveStgFree()** is not threadsafe.
- ≫ This function will fail with error code [EINVAL] if the stream file this function is operating on is a virtual volume. ≪

Related Information

- The <**Qp0lstdi.h**> file
- "QlgSaveStgFree()—Save Storage Free (using NLS-enabled path name)"—Save Storage Free (using NLS-enabled path name)
- "Save Storage Free Exit Program" on page 207

Example

See Code disclaimer information for information pertaining to code examples.

See **Qp0lGetAttr()** description for a code example that shows a call to **Qp0lSaveStgFree()** by using a procedure as the exit program. This API also shows an example of a call to **Qp0lGetAttr()**.

API introduced: V4R3

QlgSaveStgFree()—Save Storage Free (using NLS-enabled path name)

Syntax

```
#include <Qp0lstdi.h>

int QlgSaveStgFree(
  Qlg_Path_Name_T          *Path_Name,
  Qp0l_StgFree_Function_t  *UserFunction_ptr,
  void                     *Function_CtlBlk_ptr);
```

Service Program Name: QP0LLIB3
Default Public Authority: *USE
Threadsafe: Conditional; see Usage Notes for "Qp0lSaveStgFree()—Save Storage Free" on page 192.

For a description of this function and more information on the parameters, authorities required, return values, error conditions, error messages, usage notes, and related information, see "Qp0lSaveStgFree()—Save Storage Free" on page 192—Save Storage Free.

API introduced: V5R1

Save to Application (QaneSava) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	User space format name	Input	Char(8)
3	Status format name	Input	Char(8)

4	Status information	Output	Char(*)
5	Length of status information	Input	Binary(4)
6	Error code	I/O	Char(*)

Service Program Name: QANESERV
Default Public Authority: *USE
Threadsafe: No

The Save to Application (QaneSava) API enables an application to receive the save records that are generated by a save-to-save-file operation. The application defines the save operation by specifying the type of save command, and by providing the save command parameters. The API calls an exit program to transfer the save records to the application instead of to the save file.

To use the API, the application must provide the following:
- A user space that contains the required input parameter group
- An exit program

When processing the save command, the API does the following:
- Calls the exit program to indicate the start of the transfer sequence
- Submits the save command for processing
- Calls the exit program repeatedly to transfer the save records
- Calls the exit program to signal the end of the save operation
- May call the exit program to force an abnormal end to the save operation

The program that calls the API is suspended while the save operation is being processed.

Restrictions

QTEMP should not be specified for the library name on the OUTFILE or SAVACTMSGQ parameter because the save command is submitted by a prestart job running in the QSYSWRK subsystem and not in the job that called the API. Locks should not be applied to save objects that would conflict with locks applied by the save operation running in the prestart job.

Objects saved by this API can only be restored using the "Restore from Application (QaneRsta) API" on page 41 (QaneRsta) API, and only if restored to a current or a later release of the operating system from which these were saved.

The application must store the save records in the order presented, without modification, for the objects to be successfully restored.

≫ Because the save command is processed from within a prestart job, the adopted authority of the thread using the Save to Application (QaneSava) API is not available to the save command. One way to give the prestart job more authority is to use the adopted authority to swap user profiles within the application before calling the QaneSava API. ≪

Authorities and Locks

Exit Program Library Authority
        *EXECUTE

Exit Program Authority
        *EXECUTE

User Space Lock
        *SHRNUP

User Space Library Authority
     *USE

User Space Authority
     *USE

Save Command Library Authority
     *EXECUTE

Save Command Authorities
     See the save command

Saved Object Locks
     See the Back up your server topic.

Saved Object Authorities

     See Appendix D in the iSeries Security Reference book.

Required Parameter Group

Qualified user space name
     INPUT; CHAR(20)

The user space that contains all the control information for the save operation. The first 10 characters contain the user space name. The second 10 characters contain the name of the library where the user space is located.

You can use the following special values for the library name:

*CURLIB	The job's current library is used to locate the user space. If no library is specified as the current library for the job, the QGPL library is used.
*LIBL	The library list is used to locate the user space.

The actual library that is used is returned in the status information.

User space format name
     INPUT; CHAR(8)

The format name for the input parameters that are contained in the user space. For the format of the structure, see "SVRS0100 Format" on page 199.

Status format name
     INPUT; CHAR(8)

The format name for the status information returned on the API call. For the format of the structure, see "SRST0100 Format" on page 201.

Status information
     OUTPUT; CHAR(*)

The status information returned on the API call.

Length of status information
     INPUT; BINARY(4)

The length of the status information returned on the API call. The minimum length is 8 bytes.

Error code
     I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

SVRS0100 Format

This format defines the input parameter group for the API.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Length of structure
4	4	BINARY(4)	Offset to save command parameters
8	8	BINARY(4)	Length of save command parameters
12	C	BINARY(4)	Offset to application data
16	10	BINARY(4)	Length of application data
20	14	BINARY(4)	Save command type
24	18	CHAR(10)	Exit program name
34	22	CHAR(10)	Exit program library
44	2C	CHAR(8)	Target release
		CHAR(*)	Save command parameters
		CHAR(*)	Application data

Field Descriptions

Application data. Information that the application wants passed to the exit program. The content of this information is defined by the application. This field could contain information specific to the object being saved (such as the object name, size, and so forth), or it could contain the qualified name of another object that contains this information.

Exit program library. The name of the library that contains the exit program called by the API. the exit program.

Exit program name. The name of the exit program that is called by the API. See the "Save to Application Exit Program" on page 208 exit program for additional details.

Length of application data. The length of the application data. This value is passed to the exit program. This value must be set to zero if there is no application data.

Length of save command parameters. The length of the save command parameters. The maximum allowable length is 32500 bytes for save commands.

Length of structure. The length of this structure, from the start of the input parameters to the last byte of the application data.

Offset to application data. The byte offset from the beginning of the user space to the start of the application data. This value must be set to zero if there is no application data.

Offset to save command parameters. The byte offset from the beginning of the user space to the start of the save command parameters.

≫ **Save command parameters.** A character string that contains the save command parameters or save keys. These parameters are validated when the API submits the command for processing. Refer to the save commands in the Control Language (CL) information for detailed information about valid parameters. Refer to the "Save Object (QsrSave) API" on page 150 (QsrSave) API or "Save Object List (QSRSAVO) API" on page 166 (QSRSAVO) API for detailed information about valid keys. ≪

These additional restrictions apply to the save command parameters when you use this API:
- The parameters specified must be consistent with the save command type.
- The parameters must not include the save command name.
- The parameters must be separated by at least one blank character.
- Only a single library name can be used with the LIB parameter.
- ≫ The Clear (CLEAR) , Device (DEV), Save file (SAVF) and Target release (TGTRLS) parameters must not be used. These parameters are provided by the API.≪
- ≫ The Data compression (DTACPR) and Data compaction (COMPACT) parameters must not be used. These parameters are not supported by the API.≪
- ≫ The End of media option (ENDOPT), File expiration date (EXPDATE), Label (LABEL), Media definition (MEDDFN), Optical file (OPTFILE), Sequence number (SEQNBR), Starting library (STRLIB), Use optimum block (USEOPTBLK) and Volume identifier (VOL) parameters must not be used. These parameters are inconsistent with the save file operation provided by the API.≪

The following examples illustrate the save command parameters that are required for typical save scenarios:
- Example 1: Save command type 1 (SAV)

  ```
  OBJ('/*') ('/QSYS.LIB' *OMIT)

  ('/QDLS.LIB' *OMIT)
  ```
  These parameters save all objects that are not in libraries and that are not document library objects.
- Example 2: Save command type 2 (SAVOBJ)

  ```
  OBJ(FILE*) LIB(MYLIB) OBJTYPE(*FILE)
  ```
  These parameters save all files with names that start with the characters FILE* in the library named MYLIB.
- Example 3: Save command type 4 (SAVLIB)

  ```
  LIB(JOE)
  ```
  These parameters save the library named JOE.

≫ These additional restrictions apply to the command parameters when you use the Save Object (QsrSave) API or Save Object List (QSRSAVO) API. ≪
- The keys specified must be consistent with save to save file operations.
- ≫ The Clear (CLEAR), Device (DEV), Save file (SAVF) and Target release (TGTRLS) keys must not be used. These keys are provided by this API.≪
- ≫ The Data compression (DTACPR) and Data compaction (COMPACT) keys must not be used. These keys are not supported by this API.≪
- ≫ The End of media option (ENDOPT), File expiration date (EXPDATE), Label (LABEL), Media definition (MEDDFN), Optical file (OPTFILE), Sequence number (SEQNBR), Starting library (STRLIB), Use optimum block (USEOPTBLK) and Volume identifier (VOL) keys must not be used. These keys are inconsistent with the save file operation provided by the API.≪
- The starting offset for the keys is always 0 and not the offset of the save command parameters.
- All offset and integer values within the keys must be aligned on 4-byte boundaries.
- All pointers within the keys must be aligned on 16-byte boundaries.

Save command type. The type of save command that is to be processed.

1	Save (SAV) command
2	Save Object (SAVOBJ) command
3	Save Document Library Object (SAVDLO) command
4	Save Library (SAVLIB) command
5	Save Changed Object (SAVCHGOBJ) command

≫**Target release.** The release level of the operating system on which you intend to use the object being saved. The value passed in this field is ignored when the value for **Save command type** is 8 <?Pub Caret?>. The default is *CURRENT. The possible values are: ≪

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

≫Valid VxRxM0 values depend on the current version, release, and modification level, and depend on the save operation being performed. See the valid values for the TGTRLS parameter table in the Backup and Recovery book for a complete list of valid values. ≪

SRST0100 Format

This format defines the status information that is returned on the API call.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Transfer time
12	C	BINARY(4)	Transfer block size
16	10	BINARY(4)	Transfer block multiplier
20	14	BINARY(4)	Last block size
24	18	CHAR(10)	User space library used
≫34	22	CHAR(2)	Reserved
36	24	BINARY(4)	Decimal transfer time ≪

Field Descriptions

Bytes returned. The number of status information bytes returned. If the value specified in the length of status information parameter is larger than the specified status information structure, this value is set to the last byte of the returned information.

Bytes available. The number of status information bytes available for the specified status information format.

≫**Decimal transfer time.** The decimal portion of the transfer time in millionths of a second. If the value returned for Transfer time was 5 and the value returned for Decimal transfer time was 827352, then the total transfer time was 5.827352 seconds. ≪

Reserved. This field is unused and will be returned with blanks. «

Transfer block size. The number of bytes in the blocks transferred by the exit program.

Transfer block multiplier. The number of blocks successfully transferred by the exit program.

Last block size. The number of bytes in the last block transferred by the exit program.

The true transfer size of the operation is equal to the transfer block size multiplied by the transfer block multiplier plus the last block size.

Transfer time. The elapsed time, in seconds, that begins when the application calls the API, and ends when the API returns to the caller.

User space library used. The name of the user space library that is used in the API call.

Error Messages

Message ID	Error Message Text
CPF3700 E	All CPF37xx messages could be signalled. xx is from 01 to FF, excluding tape and diskette errors since the operation is a save to a save file.
CPF3800 E	All CPF38xx messages could be signalled. xx is from 01 to FF, excluding tape and diskette errors since the operation is a save to a save file.
CPF2115 E	Object &1 in &2 type *&3 damaged.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF8100 E	All CPF81xx messages could be returned. xx is from 01 to FF.
CPF9800 E	All CPF98xx messages could be signaled. xx is from 01 to FF.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.
CPFB8C0 E	Status information length for &1 API is not valid.
CPFB8C1 E	Unsupported value for &1 API.
CPFB8C2 E	Offset value for &1 API not valid. Reason &6.
CPFB8C3 E	Length value for &1 API not valid. Reason &6.
CPFB8C4 E	Unexpected condition with exit program for &1 API. Reason &6.
CPFB8C5 E	Parameter &2 specified more than once for API &1.
CPIB8C6 I	Trace data generated for API &1.
CPFB8C7 E	Unsupported value for &1 API.
CPFB8C8 E	Command syntax error detected by &1 API.
CPFB8C9 E	Command exception occurred using &1 API.
CPFB8CF E	Unexpected condition with &1 API. Reason &6.

API introduced: V4R3

<?Pub *0000027231?>

Set Cartridge Filter (QTASCTGF) API

Required Parameter Group:

1	Cartridge filter	Input	Char(*)
2	Length of cartridge filter	Input	Binary(4)
3	Format name	Input	Char(8)
4	Requested action	Input	Binary(4)
5	Error code	I/O	Char(*)

Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Set Cartridge Filter (QTASCTGF) API sets a filter that defines the cartridges that can be used by tape library devices on the system.

Authorities and Locks

The caller of this API must have *ALLOBJ and *IOSYSCFG special authorities.

Required Parameter Group

Cartridge Filter
> INPUT; CHAR(*)

> The structure that sets the cartridge filter.

Length of cartridge filter
> INPUT; BINARY(4)

> The length of the cartridge filter structure provided. The length must be set to 0 when the requested action value is 3. When the requested action value is 1, 2, or 4 the length specified must be large enough to contain the entire cartridge filter array as defined by the fields in the cartridge filter parameter.

Format name
> INPUT; CHAR(8)

> Specifies the content and format of the information being set.

> The SFTR0100 format must be used for the cartridge filter. See "SFTR0100 Format" to view the information set for this format.

Requested action
> INPUT; BINARY(4)

> The action to take with the filter provided. The possible values are:

1	The filter provided will replace the existing filter.
2	The filter provided will be appended to the existing filter.
3	The existing filter will be cleared.
4	The filter provided will be removed from the existing filter.

Error code
> I/O; CHAR(*)

> The structure in which to return error information. For the format of the structure, see Error code parameter.

SFTR0100 Format

The following table shows the information that is set for the SFTR0100 format. For more details about the fields in the following table, see "Field Descriptions" on page 204.

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Offset to array
4	4	BINARY(4)	Number of array entries
8	8	BINARY(4)	Length of array entry

Offset			
Dec	Hex	Type	Field
		Array(*) of CHAR(*)	Cartridge filter array

Field Descriptions

Cartridge filter array. An array of cartridge filters to be defined for the system. Each array entry may consist of either a specific cartridge identifier, or a generic name of a cartridge identifier. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. A generic name specifies all cartridge identifiers with names that begin with the generic prefix.

Note: The cartridge identifier is restricted to characters A through Z, zero through nine, at-sign (@), dollar-sign ($), and pound sign (#). A valid generic name contains 1 or more supported characters followed by a single asterisk (*) and blanks. The at-sign (@), dollar-sign ($), pound sign (#), and asterisk (*) characters must be provided in CCSID 37.

- @ = '7C'x
- $ = '5B'x
- # = '7B'x
- * = '5C'x

Length of array entry. This field is set to the length, in bytes, of a single cartridge filter entry. This field must be set to 6.

Number of array entries. The number of cartridge filter entries specified.

Offset to array. The offset, in bytes, to the cartridge filter array.

Error Messages

Message ID	Error Message Text
CPF222E E	&1 special authority is required.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C21 E	Format name &1 is not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C4C E	Value not valid for field &1.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF6708 E	Command ended due to error.
CPF67C9 E	An error occurred during a cartridge filter operation.
CPF67CA E	Cartridge &1 is not correct.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R3

Exit Programs

These are the Exit Programs for this category.

Restore from Application Exit Program

Required Parameter Group:

1	Operation type	Input	Binary(4)
2	Operation status	Output	Binary(4)
3	Restore data	Input	≫PTR(SPP)≪
4	Length of restore data	Input	Binary(4)
5	Restore bytes written	Output	Binary(4)
6	Qualified user space name	Input	Char(20)
7	User space format name	Input	Char(8)

The Restore from Application exit program enables an application program to provide the restore records that are required for a restore-from-save-file operation using the "Restore from Application (QaneRsta) API" on page 41 (QaneRsta) API.

The API calls the exit program once to start the transfer sequence, multiple times to transfer each block of restore records, and once to end the transfer sequence.

The API passes to the exit program the operation type, the number of restore record bytes required, the qualified name of the user space and the format name of the user space.

The exit program must return the restore data, the number of the restore record bytes retrieved, and status on the success or failure of the requested operation.

At any time following the initial call, the API could call the exit program that requires an abnormal end to the transfer sequence.

Restrictions

The exit program must provide the restore records in the order the records were saved, without modification, for the objects to be successfully restored.

Authorities and Locks

See "Restore from Application (QaneRsta) API" on page 41 (QaneRsta) API.

Required Parameter Group

Operation type
INPUT; BINARY(4)

The type of operation that the exit program is required to run.

1   Start
The exit program must use this operation type to prepare for the restore records transfer.
2   Transfer
The exit program must use this operation type to transfer (retrieve, write, and so forth) a block of restore records.
3   End
The exit program must use this operation type to end the restore records transfer.
4   Abnormal end
The exit program must use this operation type to prematurely end the restore records transfer.

Normal-operation-type order is 1 (start), 2 (transfer), 2 (transfer), ..., 2 (transfer), 3 (end).

Operation type 1 (start) is issued only once at the beginning of the restore operation before any restore records are transferred.

Operation type 2 (transfer) is issued multiple times during the restore operation as each block of restore records is required. The exit program must provide as many restore record bytes as requested, with the exception of the last block, which may not be of sufficient length.

Operation type 3 (end) is issued only once at the end of the restore operation after all restore records are transferred. The exit program must be able to handle the condition where this operation type is issued before all restore records are transferred. The exit program must handle this operation sequence as a normal condition and end the transfer sequence normally.

Operation type 4 (abnormal end) is issued only once following operation types 1 (start) or 2 (transfer), under abnormal conditions to prematurely end restore records transfer. These conditions are:

- The API detects an error with the system restore operation.
- The exit program returns an operation status of 1 (error).

Operation status
    OUTPUT; BINARY(4)

    The ending status of the requested operation.

0   Good
    The exit program must return this status value to indicate successful completion of the operation.
1   Error
    The exit program must return this status value to indicate unsuccessful completion of the operation.
2   Complete
    The exit program must use this status value instead of a status value of 0 (good)., when the last byte of the restore records has been retrieved. This indicates successful completion of operation type 2 (transfer).

Restore data
    INPUT; ≫ PTR(SPP) ≪

    ≫ A pointer to a block of restore records. ≪ The parameter is passed only on operation type 2 (transfer).

Length of restore data
    INPUT; BINARY(4)

    For operation types 1 (start), 3 (end), and 4 (abnormal end), this value is zero.

    For operation type 2 (transfer), this is the length of restore data being requested.

Restore bytes written
    OUTPUT; BINARY(4)

    For operation types 1 (start), 3 (end), and 4 (abnormal end), this value must be set to zero.

    For operation type 2 (transfer), this value must be set to the actual number of restore record bytes returned. This value must never exceed the value passed in the length of restore data parameter.

Qualified user space name
    INPUT; CHAR(20)

    The qualified user space name that is specified by the application on the call to the Restore from Application (QaneRsta) API. The first 10 characters contain the user space name. The second 10 characters contain the name of the library where the user space is located.

User space format name
    INPUT; CHAR(8)

    The user space format name that is specified by the application on the call to the Restore from Application API. For the format of the structure, see the SVRS0100 Format in the "Restore from

Application (QaneRsta) API" on page 41 (QaneRsta) API. The exit program uses the length of application data field to determine if the structure contains application data, and the offset to application data field to locate this information.

Coding Guidelines

Applications should consider the following when coding the exit program:

- The program should only return an exception for the requested operation if there has been a failure in the operation. If the program signals an escape message to the API, the system assumes there is a failure. A diagnostic message is returned to the calling program.
- The program must clean up any locks that it acquires.
- The program must handle all potential error conditions associated with its own operations (be fault tolerant).
- The program must avoid infinite looping conditions.

Exit program introduced: V4R3

Save Storage Free Exit Program

Required Parameter Group:

1	Path name pointers	Input	Char(*)
2	Return code pointer	Output	Binary(4)
3	Return value pointer	Output	Binary(4)
4	Function control block pointer	Input	Char(*)

The Save Storage Free exit program is a user-specified program that is called by **Qp0lSaveStgFree()** to save an i5/OS object of type *STMF. This exit program can be either a procedure or program.

When the Save Storage Free exit program is given control, it should save the object so it can be dynamically retrieved at a later time. The *STMF object is locked when the exit program is called to prevent changes to it until the storage free operation is complete. If the Save Storage Free exit program ends unsuccessfully, it must return a valid *errno* in the storage pointed to by the return value pointer. **Qp0lSaveStgFree()** then passes this *errno* to its caller with a minus one return code.

Storage referred to by the path name pointers or the return code pointer when the Save Storage Free exit program is called is destroyed or reused when **Qp0lSaveStgFree()** regains control.

Authorities and Locks

None.

Required Parameter Group

Path names pointers
> INPUT; CHAR(*)
>
> All of the path names to the *STMF object being storage freed. There is one path name for each link to the object. These path names are in the Qlg_Path_Name_T format and are in the UCS-2 CCSID. See Path name format for more information on this format. For information about UCS-2, see the Globalization topic.

Path Name Pointers			
Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Number of path names
4	4	CHAR(12)	Reserved
16	10	ARRAY(*)	Array of path name pointers

Array of path name pointers. Pointers to each path name that **Qp0lSaveStgFree()** found for the object identified by the path name on the call to **Qp0lSaveStgFree()**. Each path name is in the Qlg_Path_Name_T format.

Number of path names. The total number of path names that **Qp0lSaveStgFree()** found for the object identified by the caller of **Qp0lSaveStgFree()**.

Reserved. A reserved field. This field must be set to binary zero.

Return code pointer
OUTPUT; BINARY(4)

A pointer to an indicator that is returned to indicate whether the exit program was successful or whether it failed. Valid values follow:

0   The Save Storage Free exit program was successful.
-1   The Save Storage Free exit program was not successful. The Return value pointer is set to indicate the error.

Return value pointer
OUTPUT; BINARY(4)

A pointer to a valid *errno* that is returned from the exit program to identify the reason it was not successful.

Function control block pointer
INPUT; CHAR(*)

A pointer to the data that is passed to **Qp0lSaveStgFree()** on its call. **Qp0lSaveStgFree()** does not process the data that is referred to by this pointer, but passes this pointer as a parameter when it calls the exit program.

Related Information

* "Qp0lSaveStgFree()—Save Storage Free" on page 192—Save Storage Free

Exit program introduced: V4R3

Save to Application Exit Program

Required Parameter Group:

1	Operation type	Input	Binary(4)
2	Operation status	Output	Binary(4)
3	Save data	Input	» PTR(SPP) «
4	Length of save data	Input	Binary(4)
5	Save bytes read	Output	Binary(4)
6	Qualified user space name	Input	Char(20)
7	User space format name	Input	Char(8)

The Save to Application exit program enables an application program to receive the save records that are generated by a save-to-save-file operation using the "Save to Application (QaneSava) API" on page 196 (QaneSava) API.

The API calls the exit program once to start the transfer sequence, multiple times to transfer each block of save records, and once to end the transfer sequence.

The API passes to the exit program the operation type, the save data, the length of the save data, the qualified name of the user space and the format name of the user space.

The exit program must return the number of save bytes read, and status on the success or failure of the requested operation.

At any time following the initial call, the API could call the exit program that requires an abnormal end to the transfer sequence.

Restrictions

The exit program must read and store the save records in the order presented, without modification, for the objects to be successfully restored.

Authorities and Locks

See "Save to Application (QaneSava) API" on page 196 (QaneSava) API.

Required Parameter Group

Operation type
    INPUT; BINARY(4)

    The type of operation that the exit program is required to run.

1   Start
        The exit program must use this operation type to prepare for the save records transfer.
2   Transfer
        The exit program must use this operation type to transfer (copy, store, and so forth) a block of save records.
3   End
        The exit program must use this operation type to end the save records transfer.
4   Abnormal end
        The exit program must use this operation type to prematurely end the save records transfer.

        Normal-operation-type order is 1 (start), 2 (transfer), 2 (transfer), ..., 2 (transfer), 3 (end).

        Operation type 1 (start) is issued only once at the beginning of the save operation before any save records are transferred.

        Operation type 2 (transfer) is issued multiple times during the save operation as each block of save records becomes available. The exit program must read the entire block of save records.

        Operation type 3 (end) is issued only once at the end of the save operation after all save records are transferred.

        Operation type 4 (abnormal end) is issued only once following operation types 1 (start) or 2 (transfer), under abnormal conditions to prematurely end save records transfer. These conditions are:
        • The API detects an error with the system save operation.
        • The exit program returns an operation status of 1 (error).

Operation status
> OUTPUT; BINARY(4)

> The ending status of the requested operation.

0   Good
> The exit program must return this status value to indicate successful completion of the operation.

1   Error
> The exit program must return this status value to indicate unsuccessful completion of the operation.

Save data
> INPUT; » PTR(SPP) «

> » A pointer to a block of save records. « This parameter is passed only on operation type 2 (transfer).

Length of save data
> INPUT; BINARY(4)

> For operation types 1 (start), 3 (end), and 4 (abnormal end), this value is zero.

> For operation type 2 (transfer), this is the length of the block of save records.

Save bytes read
> OUTPUT; BINARY(4)

> For operation types 1 (start), 3 (end), and 4 (abnormal end), this value must be set to zero.

> For operation type 2 (transfer), the exit program must return a value that indicates the number of save record bytes successfully read. The API abnormally ends the transfer sequence if the returned value does not equal the length of save data.

Qualified user space name
> INPUT; CHAR(20)

> The qualified user space name specified by the application on the call to the Save to Application (QaneSava) API. The first 10 characters contain the user space name. The second 10 characters contain the name of the library where the user space is located.

User space format name
> INPUT; CHAR(8)

> The user space format name that is specified by the application on the call to the Save to Application API. For the format of the structure, see the SVRS0100 Format in the "Save to Application (QaneSava) API" on page 196 (QaneSava) API. The exit program uses the length of application data field to determine if the structure contains application data, and the offset to application data field to locate this information.

Coding Guidelines

Applications should consider the following when coding the exit program:

- The program should only return an exception for the requested operation if there has been a failure in the operation. If the program signals an escape message to the API, the system assumes there is a failure. A diagnostic message is returned to the calling program.
- The program must clean up any locks that it acquires.
- The program must handle all potential error conditions associated with its own operations (be fault tolerant).
- The program must prevent infinite looping conditions.

Exit program introduced: V4R3

Storage Extension Exit Program

Required Parameter Group:

1	Object description information	Input	Char(*)
2	Control value information	Output	Char(*)

QSYSINC Member Name: ETASTGEX
Exit Point Name: QIBM_QTA_STOR_EX400
Exit Point Format Name: EX400200, EX400300

The Storage Extension exit program provides the capability of restoring the entire object using a storage extension, that is, restoring objects that were saved using *FREE, or freed through an application programming interface. (*DOC and *STMF files are freed through an API, not through save using *FREE.)

Note: To use this exit program, you need the Media and Storage Extension feature of the i5/OS.

If there is any program registered against exit point format name EX400200 of this exit point, then any programs registered against exit point format name EX400300 will not be called. Therefore, if you are installing your application and you are registering it against exit point format EX400300, verify that no programs are registered against exit point format name EX400200. If there are any, notify the user that it needs to be disabled before the application will work. Do not simply deregister programs from exit point format name EX400200 when installing your application because it may impact other applications.

Storage extension refers to those objects (and the CL commands that refer to those objects) saved from disk using the *FREE option on the storage parameter. These saved objects free disk space by storing a copy of the entire object and keeping only the object headers on disk. Currently, only file, document, and stream objects are supported.

Exit Point Format EX400200

Objects may be scheduled to be saved from disk when they are not referred to for a specified amount of time. When the objects are saved, the object data is saved and the object headers remain on disk. When this object is referred to, the operating system calls the exit program for object restoration through the registration facility. (For information about registering an exit point with the registration facility and adding an exit program to an exit point, Registration Facility APIs. The exit point format EX400200 supports only one exit program.)

When the user exit program is given control, it verifies that the object was saved. If the exit program has the object saved and wants to restore it, the exit program restores the object data and returns a control value to the i5/OS operating system indicating that the object was restored through the control value information. If the exit program does not have the object saved, it returns a control value to i5/OS indicating that the object was not restored through the control value information.

Exit Point Format EX400300

When an object is determined to be saved using *FREE, each program that is registered against exit point format name EX400300 will be called (as long as no programs are registered against EX400200) with an indicator that it is asking for a date/time stamp of the most recent version of the object that the exit program has.

After all programs are called, the exit program that specified the most recent date/time stamp will be called again with the indicator to restore the object.

After the user exit program is given control and restores the object that was suspended, it should return the control value to the i5/OS operating system indicating that the object was restored through the control value information.

Authorities and Locks

None.

Required Parameter Group

Object description information
  INPUT; CHAR(*)

  Information about the object that the exit program will attempt to restore from storage extension. For details, see "Format of Object Description Information (EX400200,EX400300)."

Control value information
  OUTPUT; CHAR(*)

  Information about whether the exit program restored the object requested or did not have the object stored in storage extension. For details, see "Format of Control Value Information" on page 214.

Format of Object Description Information (EX400200,EX400300)

The following table shows the format of the object description information. For a description of the fields in this format, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Length of object description information
4	4	BINARY(4)	Length of control value information
8	8	CHAR(10)	Object name
18	12	CHAR(10)	Object library name
28	1C	CHAR(10)	Object type
38	26	CHAR(10)	Member name
48	30	CHAR(10)	Job name
58	3A	CHAR(10)	User name
68	44	CHAR(6)	Job number
74	4A	CHAR(2)	Reserved
76	4C	BINARY(4)	Offset to path name structure
80	50	BINARY(4)	Length of path name structure
84	54	CHAR(10)	Request type
94	5E	CHAR(13)	Date/time stamp
		CHAR(*)	Path name structure

Field Descriptions

Date/time stamp. The most recent date/time stamp that the other exit programs have specified as their most recent copy of the suspended object. If this is the first exit program being called, or no other exit

program has a copy of the suspended object to be restored, then this field will be set to blanks. This field will be blanks when passed for exit format EX400200. This field will be of the following CYYMMDDHHMMSS format:

C	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
YYMMDD	The date (year, month, day format).
HHMMSS	The time (hours, minutes, seconds format).

Job name. The job name.

Job number. The job number associated with the job name and user identifier.

Length of control value information. The length, in bytes, of the control value information.

Length of the path name structure. The length, in bytes, of the path name structure. This field will be set to zero if the object does not have a path name structure passed.

Length of object description information. The length, in bytes, of the object description information.

Member name. The member within the file that caused the exception.

Object library name. The library name of the object being referred to. The special value is:

*PATH	The path name structure will contain the object information.

Object name. The name of the object that is being referred to and that causes an exception. The user exit program checks if it has this object saved to storage extension. The special value is:

*PATH	The path name structure will contain the object information.

Object type. The standard object types known to the system. Currently, only the following object types are supported:

*FILE	File object (object name and library fields will contain object name information).
*DOC	Document object (path name structure will contain object name information).
*STMF	Stream file object (path name structure will contain object name information).

Offset to path name structure. The offset, in bytes, to the path name structure that is passed containing object pathname and translation information. This field will be set to zero if the object does not have an path name structure.

Path name structure. The path name structure and translation information for the suspended object. The path name structure contains information such as CCSID, country or region, and language. For more information on this structure, see Path Name Format.

Request type. The type of request to the exit program from the operating system. This field will always be *RESTORE for exit format EX400200. Possible values are:

*RESTORE	The exit program is getting called to restore the object.
*DATETIME	The exit program is getting called to return the latest date/time stamp of the most recent save operation of the suspended object. Note that i5/OS does not restrict the called exit program from actually restoring the object when called for a date/time stamp, but it will only be a degradation in performance due to an extra restore of the object.

Reserved. An unused field.

User name. The user identifier of the caller.

Format of Control Value Information

The following table shows the format of the control value information. For a description of the fields in this format, see "Control Value Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	CHAR(1)	Object restoration information
1	1	CHAR(13)	Date/time stamp

Control Value Field Descriptions

Date/time stamp. This field should be set by the exit program when the request type specified in the "Format of Object Description Information (EX400200,EX400300)" on page 212 is *DATETIME. This field is used by the operating system to determine which registered exit program will be called again to restore the object. The field is only used when programs registered under exit point format EX400300 are called. The determination is based on which exit program indicates the most recent copy of the suspended object. This field will be of the following CYYMMDDHHMMSS format:

C	Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
YYMMDD	The date (year, month, day format).
HHMMSS	The time (hours, minutes, seconds format).

Object restoration information. Whether or not the object was successfully restored or whether the exception should be resignaled. If this field contains a value that is not valid, the value is ignored and message CPD6705 is issued. The default for this field is 0. Valid values are:

0	The object has not been restored or was not asked to be restored through the request type field. **Note:** This field should always be left by the exit program as 0 when the request type specified in the "Format of Object Description Information (EX400200,EX400300)" on page 212 is *DATETIME.
1	The object has been restored. **Note:** If the user exit program specifies a 1 for this field and it did not attempt to (or successfully) restore the entire object, message CPD6704 is signaled.

Error Messages

Message ID	Error Message Text
CPD6704 D	Error detected using program &1 in &2.
CPD6705 D	Incorrect user exit control value specified.

Exit program introduced: V3R1

Tape Management Exit Program

Required Parameter Group:

1	Exit description information	Input	Char(*)
2	Label information	Input	Char(*)
3	Operational information	Input	Char(*)
4	Control value information	Output	Char(*)

QSYSINC Member Name: ETATAPMG
Exit Point Name: QIBM_QTA_TAPE_TMS
Exit Point Format Name: TMS00200
Threadsafe: No

The Tape Management exit program allows a tape management system to monitor and control the use of volumes and devices used by the operating system for most tape operations. The exit program is given control at certain points during tape and library processing.

Note: To use this exit program, you need the Media and Storage Extension feature of the operating system.

The exit program is not given control when:
- The system is being installed
- The tape job is a dedicated service tools (DST) tape job
- The job is ending

The operating system handles the setup, calling, and response processing of the user exit program as specified through the registration facility. (For information about registering an exit point with the registration facility and adding an exit program to an exit point, Registration Facility APIs. This exit point supports only one exit program.)

Authorities and Locks

None.

Required Parameter Group

Exit description information
        INPUT; CHAR(*)

        A description of the exit point. For a description of the format, see "Format of Exit Description Information" on page 216.

Label information
        INPUT; CHAR(*)

        The current volume label and the last header label or trailer label that was written or read. For a description of the format, see "Format of Label Information" on page 218.

Operational information
        INPUT; CHAR(*)

        Information about the tape operation at the time the exit program is called. For a description of the format, see "Format of Operational Information" on page 218.

Control value information
        OUTPUT; CHAR(*)

Information to control the tape operation being performed. This format is set by i5/OS and may be changed by the exit programs to control tape processing. For a description of the format, see "Format of Control Value Information" on page 226.

Format of Exit Description Information

The following table shows the format of the exit description information. For a description of each field, see "Field Descriptions."

Offset			
Dec	Hex	Type	Field
0	0	BINARY(4)	Length of exit description information
4	4	CHAR(1)	Tape position exit type
5	4	CHAR(1)	Tape library device exit type

Field Descriptions

Length of exit description information. The length, in bytes, of the exit description information.

Tape library device exit type. An identifier that indicates to the exit program the type of library processing occurring in the tape library device. The values are:

0   Ignore
    Use the value specified in the tape position exit type field.
1   Addition
    This exit type occurs immediately after the cartridge identifier is added to a tape library device using the Add Tape Cartridge (ADDTAPCTG) command.
2   Removal
    This exit type occurs immediately before the cartridge identifier is removed from a tape library device using the Remove Tape Cartridge (RMVTAPCTG) command.
3   Category
    This exit type occurs immediately before the cartridge identifier has its category changed from one category to another using the Change Tape Cartridge (CHGTAPCTG) command.
4   Mismatch
    This exit type occurs whenever a mismatch is found between a cartridge identifier and the volume identifier on the tape cartridge.
5   Mount failure
    This exit type occurs when a cartridge failed to be mounted. It gives the tape management program an opportunity to choose a different cartridge.
6   Unload exit
    This exit type occurs after taking the option reject and unload. It gives the tape management program an opportunity to choose the next cartridge to be mounted.
7   Mount category exit
    This exit type occurs before a category is mounted through the use of the Set Tape Category (SETTAPCGY) command with *MOUNTED specified for the option parameter. This exit type allows the tape management program the opportunity to reject the SETTAPCGY command.
8   Demount category exit
    This exit type occurs before a category is demounted through the use of the Set Tape Category (SETTAPCGY) command with *DEMOUNTED specified for the option parameter. This exit type allows the tape management program the opportunity to reject the SETTAPCGY command.
9   Inventory success exit
    This exit type occurs after a successful inventory has been received from the tape library device.

Tape position exit type. An identifier that indicates to the exit program a reference to or the position of the tape. The values are:

0   Ignore
      Use the value specified in the tape library device exit type field.
1   Start of file (SOF)
      This exit type occurs late in the open tape file processing but before any steps related to mounting the first
      tape volume. This exit type provides information concerning the open processing and allows the exit program
      to select the first tape volume that will be read from or written to.
2   Start of volume (SOV)
      This exit type occurs immediately after the volume label is read unless an initialize operation is being done.
      The purpose is to enable acceptance or rejection of a tape volume and to allow the exit program to record the
      volume actually used. This exit type occurs once for each volume processed. When an initialize operation is
      being done, this exit type occurs before the volume label is written. The current volume identifier (if known)
      is the volume identifier before the initialize operation and the next volume identifier is the new volume
      identifier.
3   Start of file section (SOS)
      This exit type occurs immediately after the HDR2 file header label is read on input or immediately before the
      HDR2 label is written on output. Its purpose is to enable acceptance or rejection of a tape that was previously
      accepted at start-of-volume exit type. This exit type occurs once for each file processed.
4   End of file section (EOS)
      This exit type occurs immediately after the EOV2 end-of-volume label is read on input or immediately after
      the EOV2 label is written on output. This exit type allows the exit program to select the next tape volume to
      be read from or written to. This exit type occurs once for each volume processed except for the last volume.
      (This exit type does not occur for single volume processes.)
5   End of file (EOF)
      This exit type occurs immediately after the EOF2 end-of-file label has been read on input or immediately after
      the EOF2 label is written on output. Its purpose is to inform the exit program that tape processing is
      complete.
6   Message
      This exit type occurs immediately before a message is sent by the tape manager. The exit type informs the exit
      program that a message will be sent. If the message is an inquiry message, the valid responses accepted by
      the message handler may be the same as those that the i5/OS program accepts from the tape management
      system.
7   End position
      This exit type occurs immediately before an end positioning occurs. End positioning refers to whether the
      tape is rewound, unloaded, or left in leave processing. (**Leave processing** refers to the use of
      ENDOPT(*LEAVE) on a tape command.) The exit type is driven by the End Option (ENDOPT) parameter on
      all tape commands. These values could be ENDOPT(*LEAVE), ENDOPT(*REWIND), or ENDOPT(*UNLOAD).
      User-specified values cannot be overridden in error scenarios because tape volumes are always rewound in
      error situations.
8   Command exit
      This exit type occurs before the start-of-file exit type and is designed to allow tape management systems the
      ability to choose values that need to be known before any operation is performed on the tape device. It is
      only enabled as an exit type when options can be changed.

Examples of Exit Calls

This example shows the sequence and the tape position exit types that result from saving a library object
(SAVLIB command) to one tape:

- Command (CMD)
- Start of file (SOF)
- Start of volume (SOV)
- Start of file section (SOS)

  After this exit type, the data file is written.

- End of file (EOF)
- End position

This example shows the sequence and the tape position exit types that result from saving one library object to two tapes:

- Start of command (CMD)
- Start of file (SOF)
- Start of volume (SOV)
- Start of file section (SOS)

  After this exit type, the first part of the data file is written.
- End of file section (EOS)

  After this exit type, a new tape volume is requested.
- Start of volume (SOV)
- Start of file section (SOS)

  After this exit type, the next part of the data file is written to the second tape.
- End of file (EOF)
- End position

Format of Label Information

The following table shows the format of the label information. For a description of each field, see "Field Descriptions."

| Offset | | | |
Dec	Hex	Type	Field
0	0	BINARY(4)	Length of label information
4	4	CHAR(80)	Current volume label
84	54	CHAR(80)	Last processed HDR1 or TRL1 label
164	A4	CHAR(80)	Last processed HDR2 or TRL2 label

Field Descriptions

Current volume label. The volume label currently being processed. If the tape position exit type is SOF or if the tape library device exit type is addition, removal, category, or mismatch, this field contains blanks. This field also contains blanks if a nonlabeled tape is being processed.

Last processed HDR1 or TRL1 label. The HDR1 file header label or TRL1 trailer label that was last encountered. If the tape position exit type is SOF or SOV or if the tape library device exit type is addition, removal, category, or mismatch, this field contains blanks. This field also contains blanks if a nonlabeled tape is being processed.

Last processed HDR2 or TRL2 label. The HDR2 file header label or TRL2 trailer label that was last encountered. If the tape position exit type is SOF or SOV or if the tape library device exit type is addition, removal, category, or mismatch, this field contains blanks. This field also contains blanks if a nonlabeled tape is being processed.

Length of label information. The length, in bytes, of the label information.

Format of Operational Information

| Offset | | | |
Dec	Hex	Type	Field
0	0	BINARY(4)	Length of operation information

Offset		Type	Field
Dec	Hex		
4	4	BINARY(4)	Length of control value information
8	8	CHAR(1)	Tape operation
9	9	CHAR(17)	Data file label
26	1A	CHAR(10)	Tape device file name
36	24	CHAR(10)	Tape device file library name
46	2E	CHAR(10)	Current device name
56	38	CHAR(6)	Current volume identifier
62	3E	CHAR(10)	Next device name
72	48	CHAR(6)	Next volume identifier
78	4E	CHAR(4)	Current device type
82	52	CHAR(10)	Current tape density
92	5C	CHAR(1)	Data check on write
93	5D	CHAR(10)	Next tape density
103	67	CHAR(1)	Tape device ready status
104	68	CHAR(1)	Tape volume initialize status
105	69	CHAR(1)	Initialize new volume label
106	6A	CHAR(32)	Logical block identifier
138	8A	CHAR(6)	Cartridge identifier
144	90	CHAR(10)	Category name
154	9A	CHAR(8)	Category system name
162	A2	CHAR(1)	Mismatch status
163	A3	CHAR(10)	Library device name
173	AD	CHAR(1)	Library device status
174	AE	CHAR(1)	Library device mode
175	AF	CHAR(1)	System restricted state status
176	B0	CHAR(1)	Tape volume write protection status
177	B1	CHAR(7)	Message identifier
184	B8	CHAR(10)	Message type
194	C2	CHAR(10)	Message queue or program name
204	CC	CHAR(10)	Message queue or program library name
214	D6	CHAR(4)	Message destination
218	DA	CHAR(1)	Volume list status
219	DB	BINARY(4)	Offset of replacement text
223	DF	BINARY(4)	Length of replacement text
227	E3	CHAR(1)	Generated cartridge ID status
228	E4	CHAR(1)	Sequence number change allowed
229	E5	CHAR(1)	End position
230	E6	CHAR(10)	File sequence number
240	F0	CHAR(10)	Aggregate volume sequence number
250	FA	CHAR(10)	Media library tape resource name

Offset			
Dec	**Hex**	**Type**	**Field**
260	104	CHAR(4)	Media library tape resource type
264	108	CHAR(4)	Media library tape resource model
268	10C	CHAR(1)	Generated cartridge identifier
269	10D	CHAR(10)	Session identifier
279	117	CHAR(150)	Cartridge densities
429	1AD	CHAR(26)	Qualified job name
455	1C7	CHAR(4)	Reserved
459	1CB	CHAR(1)	Type of close operation
460	1CC	CHAR(10)	Command name
470	1D6	CHAR(1)	Message recoverable flag
471	1D7	CHAR(1)	Output extend processing
472	1D8	CHAR(10)	User sequence number
482	1E2	CHAR(6)	User expiration date
» 488	1E8	CHAR(1)	Virtual device «
» 489	1E9	CHAR(1)	WORM cartridge «
*	*	CHAR(*)	Message replacement text (The offset to this field is stored in the Offset of replacement text field)

Field Descriptions

Aggregate volume sequence number. The aggregate volume sequence number from the label information. The numeric value is right-justified with leading zeros or blanks.

Cartridge densities. The densities that are supported by the cartridge. Up to 15 densities or formats are supported by the cartridge. Each density or format is 10 characters in length. For example, if only one density exists, the first 10 bytes of this field are the character representation of the density or format and the last 140 bytes are blanks. This field is set at the start-of-volume exit type and is blank at all other exit types. This field is blank for devices or cartridges that do not support special cartridge-checking capabilities.

Cartridge identifier. The cartridge identifier of the tape cartridge. If the tape library device has a scanner, the cartridge identifier is the external bar-code identifier. If the tape library device does not have a scanner, the cartridge identifier is the logical volume identifier. This field is blank if the device is not in a tape library device or when the tape position exit type field is SOF.

Category name. The category name that the cartridge identifier is being changed to. This field is blank if the tape library device exit type field is not addition or category.

Category system name. The category system name that is the primary owner of the category name. This field is blank when the category is *INSERT, *EJECT, or *SHARE400.

Command name. The name of the command being run. If this field is blanks, then the command does not support passing the command name to the Media and Storage Extension structures at this exit point. Check Tape (CHKTAP), Duplicate Tape (DUPTAP), and Initialize Tape (INZTAP) commands pass the command name for all exit types.

Current device name. The name of the tape device being used at the time the exit point is reached. This field is blank if the tape library device exit type field is addition, removal, or category.

Current device type. The device type of the current tape device. This field is blank if the tape library device exit type field is addition, removal, or category.

Current tape density. The density of the tape reel or cartridge on the current tape device. This field is blank if the tape library device exit type field is addition, removal, or category.

Current volume identifier. The name of the expected volume to be used during the tape operation, not necessarily the loaded volume. i5/OS issues an inquiry message when the expected volume is different from the loaded volume. This field is blank if the tape library device exit type field is addition, removal, or category.

Data check on write. Whether a permanent write data check occurred. Valid values are:

0   A permanent write data check has not occurred.
1   A permanent write data check occurred and end-of-volume labels are being written before the end of tape.

Data file label. The tape data file being processed. This field is blank if the tape library device exit type field is addition, removal, or category.

End position. The type of end position that is occurring for this close operation. This exit type is designed for applications that track the end positioning of cartridges. It becomes important for media library devices where *LEAVE ties up a device for that cartridge being left in leave processing. This field is blank at all exit points other than the end position exit type. Valid values follow:

0   *REWIND
1   *UNLOAD
2   *LEAVE

File sequence number. The file sequence number from the label information.

Generated cartridge identifier. Whether the cartridge identifier is generated. Generated identifiers include BLKxxx (where xxx are the characters 0-9) for blank or new tapes, IMPxxx for tapes in the convenience (import) station, NLTxxx for nonstandard labeled tapes, CLNxxx for cleaning tapes, and ERRxxx for tape cartridges in error. This field is blank when no cartridge identifier is passed in the operational information format in the cartridge identifier field. Valid values follow:

0   The cartridge identifier is not generated.
1   The cartridge identifier is generated.

Generated cartridge ID status. Whether the media library device has a bar-code reader. If the media library device does not have a bar-code reader, then the field indicates if the cartridge identifiers are generated by the system or if the logical volume identifiers are used as the cartridge identifiers. This field is blank at all exit types except inventory success exit. Valid values follow:

0   The media library device has a bar-code reader.
1   The media library device does not have a bar-code reader, and the logical volume identifiers will be used as the cartridge identifiers (that is, the device description has *VOLID specified for the GENCTGID parameter).
2   The media library device does not have a bar-code reader, and the system will generate the cartridge identifiers (that is, the device description has *SYSGEN for the GENCTGID parameter).

Initialize new volume label. Whether a new volume label initializes immediately. This field is blank if the tape position exit type field is not SOV. Valid values are:

0   An SOV tape position exit type after the volume label on tape is read
1   An SOV tape position exit type before the new volume label is written

Length of control value information. The length, bytes, of the control value information.

Length of operational information. The length, in bytes, of the operational information.

Length of replacement text. The length, in bytes, of the replacement text for the message exit type.

Library device mode. Whether the library device is in library mode. Valid values are as follows:

0   The library device is not in library mode.
1   The library device is in library mode.

Library device name. The name of the tape library device being used at the time the exit point is reached. This field is blank if a tape library device is not being used.

Library device status. Whether the device is in a library device. Valid values are as follows:

0   The tape device is not in a library device.
1   The tape device is in a library device.

Logical block identifier. The current tape position. This refers to the logical position of the data buffer rather than the physical position of the media. This field is passed as part of the SOS tape position exit type for tapes that support positioning by logical block identifier on standard labeled (*SL) tapes opened for output. This field is zero at all other exit types.

Media library tape resource name. For a media library device, the resource name of the tape device being used. If no device is allocated to the process for use, the field is blank.

Media library tape resource model. The model number of the tape resource name.

Media library tape resource type. The type number of the tape resource name.

Message destination. The destination of the message to be signalled by the operating system in relation to the program name specified in the message queue or program name field. Valid values are as follows:

PREV   The message is sent to the program previous to the one specified.
SAME   The message is sent to the program specified.

Message identifier. The identifier of the message that is signalled.

Message queue or program name. The name of the message queue or program to which the message will be sent.

Message queue or program library name. The name of the library in which the message queue or program will reside.

Message recoverable flag. Indicates if the message is recoverable. This field is valid only for the message exit. It is blank for all other exit types. Valid values are as follows:

0   The message is not recoverable.
1   The message is recoverable.

Message replacement text. The replacement text for the message identifier on the message exit type. The offset to this field and the length of this field are also contained in the operational information format.

Message type. The type of message that will be signalled. Valid values are as follows:

ESCP   Escape message
INQ   Inquiry message

Mismatch status. Whether a mismatch occurred when using the Add Tape Cartridge (ADDTAPCTG) command. This field is blank if the tape library device exit type is not mismatch or if the device is not in a tape library device. Valid values are:

0   A mismatch did not occur when using the command.
1   A mismatch occurred when using the command.

Next device name. The next device name in the list of devices. If only one device has been specified, the current and next device names are the same. This field is blank if the tape library device exit type is addition, removal, or category.

Next tape density. The new density when a reel or cartridge is initialized. This field is blank if the tape library device exit type is addition, removal, or category.

Next volume identifier. The next volume label in the list of labels. Blanks indicate that the volume identifier list is used up. This field is blank if the tape library device exit type is addition, removal, or category.

Offset of replacement text. The offset, in bytes, of the replacement text for the message exit type.

Output extend processing. Indicates when a tape file is opened for output with extend processing. This flag will only be set when a tape file is opened for output. Valid values are:

blank   The file is opened for input.
0   The file is opened for output.
1   The file is opened for output with extend processing.

Qualified job name. The qualified job name that forced the tape function and exit types.

Reserved. An ignored field.

Session identifier. An identifier to the tape management system of the session identifier that is associated with this request. Each session in a media library device has its own unique session identifier. An exception is a mounted category request, which has a session identifier assigned from the SETTAPCGY OPTION(*MOUNTED) to SETTAPCGY OPTION(*DEMOUNTED); this session identifier is used for each command that specifies VOL(*MOUNTED) while a category is mounted. This field is set to blanks for stand alone devices.

Sequence number change allowed. Whether the tape management system is allowed to change the user-specified sequence number. Valid values follow:

blank   No change to the user-specified sequence number value is allowed.

0	No change to the user-specified sequence number value is allowed because of the tape positioning or tape file status (that is, end of volume).
1	A change to the user-specified sequence number is allowed. Valid sequence numbers include 1, *END, or one greater than the last sequence number on the tape. This value is valid only for output operations.
2	A change to the user-specified sequence number is allowed.
	For output operations, the valid new sequence number values include 1, any existing sequence number, *END, or one greater than the last sequence number on the tape.
	For input operations, the valid new sequence number values include any existing sequence number.

When a change is allowed, the user-specified sequence number is placed in both the sequence number and large sequence number fields of the control value information. Either the sequence number or large sequence number fields of the control value information may be changed to a new value by the tape management system.

Note: For user-specified sequence numbers greater than 999 999, the value will appear only in the large sequence number field.

Note: The tape management system is not allowed to change the user-specified sequence number for input operations for the CHKTAP, DMPTAP, DSPTAP, and DUPTAP commands.

System restricted state status. Whether the system is in restricted state. Valid values are as follows:

| 0 | The system is not in a restricted state. |
| 1 | The system is in a restricted state. |

Tape device file library name. The name of the library that contains the tape device file. This field is blank if the tape library device exit type is addition, removal, or category.

Tape device file name. The name of the tape device file for this tape device. This field is blank if the tape library device exit type is addition, removal, or category.

Tape device ready status. Whether the tape device is ready. Valid values are:

| 0 | The tape device is not ready |
| 1 | The tape device is ready |

Tape operation. An indicator of how the tape file was opened. Valid values are:

0	The tape file is open for input.
1	The tape file is open for output.
2	No tape file is open.

Tape volume initialize status. The identifier that determines the type of checking for active files during the initialize operation. This field is blank when it is not an initialize operation. Valid values are:

0	Initialize operation and check for active file is *NO.
1	Initialize operation and check for active files is *YES.
2	Initialize operation and check for active files is *FIRST.

Tape volume write protection status. Whether the tape volume in use is write-protected. Valid values are as follows:

0	The tape volume is not write-protected.
1	The tape volume is write-protected.

Type of close operation. The type of close operation. If the exit type is not an end-of-file exit type, the field is blank. Valid values follow:

blank	Not an end-of-file exit type
1	Temporary close (save or restore operation closes between files)
2	Permanent close (operation has completed)
3	Normal termination (operation is being ended normally)
4	Abnormal termination (operation is being ended abnormally)

User expiration date. The expiration date specified by the user when the tape file was opened.

This field uses the Julian format with a leading century digit. It is of the form CYYDDD, where C represents the thousands and hundreds of the year value (19 is blank, 20 is 0, and 21 is 1), YY represents the year (00-99), and DDD represents the day of the year (001-366). For example, 1 February 1972 is represented as ' 72032' and 1 February 2072 is represented as '072032'.

The value *PERM indicates that the tape file is a permanent tape file.

The user expiration date will be blanks when the file is open for input or if the expiration date is *NONE.

User sequence number. The sequence number specified by the user when the tape file was opened. The sequence number will be blank when a tape file is opened for accessing a tape device but no input/output operations are to be issued or no sequence number is specified. Valid values are:

FIRST	First sequence on the tape (only valid for input)
NEXT	Next sequence on the tape (only valid for input)
END	Append to end of the tape (only valid for output)
number	The specified sequence on the tape (valid for input and output)

» Virtual device. Indicates whether or not the device is a virtual device. Valid values are:

0	The device is not a virtual device.
1	The device is a virtual device. «

Volume list status. Whether *MOUNTED is specified for the volume (VOL) parameter. This field is set at SOF and SOV. This field allows tape management to control the monitoring of the Set Tape Category (SETTAPCGY) command for occurrences of *MOUNTED specified for the VOL parameter. This field is blank if the tape processing exit type is not start of file (SOF) and not start of volume (SOV). Valid values are as follows:

0	VOL(*MOUNTED) is not specified.
1	VOL(*MOUNTED) is specified.

» WORM cartridge Specifies whether the current cartridge is a write-once-read-many (WORM) cartridge or a read-write cartridge. This field is valid when the tape processing exit type is start of volume (SOV), start of file section (SOS), end of file section (EOS), or end of file (EOF). It will be blank for all other exit types. Valid values are as follows:

0   The current cartridge is a read-write cartridge.
1   The current cartridge is a write-once-read-many cartridge. «

Format of Control Value Information

Offset			
Dec	Hex	Type	Field
0	0	CHAR(1)	Volume acceptance
1	1	CHAR(6)	Volume to be used
7	7	CHAR(6)	File expiration date
13	D	CHAR(1)	Character code conversion
14	E	CHAR(1)	Allow ignore response
15	F	CHAR(1)	Issue active file messages
16	10	CHAR(1)	Issue mount next tape message
17	11	CHAR(32)	Logical block identifier
49	31	CHAR(1)	Allow category change
50	32	CHAR(1)	Allow removal
51	33	CHAR(1)	Mismatch acceptance
52	34	CHAR(1)	Automatic ADDTAPCTG
53	35	CHAR(1)	Message response
54	36	CHAR(1)	Allow mount category
55	37	CHAR(1)	Allow demount category
56	38	CHAR(6)	Sequence number
62	3E	CHAR(10)	Category name
72	48	CHAR(8)	Category system name
80	50	CHAR(10)	Large sequence number
90	5A	CHAR(1)	Allow logical block identifier support
91	5B	CHAR(1)	Allow cartridge search
92	5C	CHAR(1)	End position
93	5D	CHAR(10)	Density or format
103	67	CHAR(1)	Use optimum block size
104	68	CHAR(1)	Duplicate file
» 105	69	CHAR(10)	Image catalog name «
» 115	73	CHAR(1)	Suppress auto-generate of virtual tape volume «

Field Descriptions

All of these control values have a default at the exit types in which the exit program can change them. For exit types that values cannot be changed, the values are set to blanks.

Allow cartridge search. Whether to allow cartridge searching for non-bar-code media library devices when the cartridge that is specified in the VOL parameter of a command is not found. If the system is allowed to use cartridge searching, the system loads tapes and searches for a logical volume identifier to

match the requested volume. The cartridges that the system loads and searches are the cartridges with generated identifiers and unknown logical volume identifiers. Cartridges with generated identifiers such as NLTxxx (nonlabeled), BLKxxx (blank), or ERRxxx (error) are not used in cartridge searching by the system.

If cartridge searching is allowed, the system may load and unload the convenience station tape, which causes the tape to no longer be recognized by the device. This field can only be changed at the start-of-file exit type. At all other exit types, the default value is blank. Valid values follow:

0   Disallow system cartridge searching
1   Allow system cartridge searching

Allow category change. Whether to allow a change of category for the cartridge identifier. You can change this value when the tape library device exit type field is category. The default is 1 when the tape library device exit type is category. Valid values are:

0   The category for the cartridge identifier cannot be changed.
1   The category for the cartridge identifier can be changed.

Allow demount category. Whether to allow a demount category. You can change this value when the tape library device exit type field is demount category. The default is 1 when the tape library device exit type is demount category. Valid values are as follows:

0   Disallow the demount category operation.
1   Allow the demount category operation.

Allow ignore response. Whether to allow an ignore response to a mount message. If the field contains a value that is not valid, it is ignored and message CPF4067 is issued. You can change the value when the tape position exit type field is SOV. The default is 0 when the tape position exit type is SOV. Valid values are:

0   An ignore response is allowed from the mount next tape messages. The ignore response is permitted as normal on the mount next tape messages.
1   An ignore response is not allowed from the mount next tape messages.

Allow logical block identifier support. Whether to allow logical block identifier support for this device. If a tape management system responds at the start-of-file exit type, this value allows devices that do not support logical block identifiers to work with i5/OS code. Devices that are emulating i5/OS supported devices may not support logical block identifiers but still report because of emulation that the device does support it. Valid values follow:

0   Disallow logical block identifier support.
1   Allow logical block identifier support.

Allow mount category. Whether to allow a mount category. You can change this value when the tape library device exit type field is mount category. The default is 1 when the tape library device exit type is mount category. Valid values are as follows:

0   Disallow the mount category operation.
1   Allow the mount category operation.

Allow removal. Whether to allow the removal of the cartridge identifier. You can change the value when the tape library device exit type field is removal. The default is 1 when the tape library device exit type is removal. Valid values are:

0   The cartridge identifier cannot be removed from the tape library device.
1   The cartridge identifier can be removed from the tape library device.

Automatic ADDTAPCTG. Whether to allow the tape management system the ability to automatically add the cartridge to a usable category when the exit type field is mount failure and the cartridge is currently in the *INSERT category. (For more information about usable categories, see the Add Tape Cartridge (ADDTAPCTG) command in the Control Language (CL) topic.) You can change this value when the tape library device exit type field is mount failure. The default is 0 when the tape library device exit type is mount failure. Valid values are:

0   The cartridge identifier is not added to a usable category.
1   The cartridge identifier is added to the *NOSHARE category, and the tape processing continues.
2   The cartridge identifier is added to the *SHARE400 category, and the tape processing continues.
3   The cartridge identifier is added to the *IPL category, and the tape processing continues.
4   The cartridge identifier is added to the *NL category, and the tape processing continues.
5   The cartridge identifier is added to the *CNV category, and the tape processing continues.
6   The cartridge identifier is added to the category name and to the category system name fields that are specified in the control value information format.

Category name. If a valid category name and category system name are specified, the cartridge is changed to that category. This automatic Change Tape Cartridge (CHGTAPCTG) command does not force a change exit type and can only be specified for media library devices and at the start-of-volume exit type. If you specify option 6 for the automatic tape cartridge field at the mount failure exit type, this field is also allowed. The field is set to blanks at all other exits types. You cannot change to the system-supplied categories *CNV and *SYSGEN.

Category system name. If you specify a valid category name and category system name, the cartridge is changed to that category. This automatic Change Tape Cartridge (CHGTAPCTG) command does not force a change exit type and can only be specified for media library devices and at the start-of-volume exit type. If you specify option 6 for the automatic tape cartridge field at the mount failure exit type, this field is also allowed. The field is set to blanks at all other exit types. You cannot change to the system-supplied categories *CNV and *SYSGEN.

Character code conversion. Whether to convert character code from ASCII to EBCDIC for data written on the tape. If the field contains a value that is not valid, it is ignored and message CPF4067 is issued. You can change this value when the tape position exit type field is SOF and the tape operation field is 1. The default is 0 when the tape position exit type is SOF and the tape operation field is 1.

0   Convert ASCII data to EBCDIC data when processing the data file
1   Retain ASCII data

Density or format. This field allows the tape management system to change the density or format of the tape volume. When the initialize new volume label (operational information format) is set to 1, which indicates that the new tape volume is written immediately, the user-specified density at the start-of-volume exit type is used. The field must be set to one of the valid cartridge densities that are listed in the operational information format. The field is blank when it cannot be changed.

Duplicate File. This field allows the tape management system to select which files are duplicated. You can change this value when the tape position exit type field is SOS and the tape operation field is 0 for the DUPTAP command. The default is to duplicate the file.

0	Do not duplicate the file
1	Duplicate the file

End position. This field allows the tape management system to change the end positioning that was specified by the user. The value can be changed at the end positioning exit type. This field is blank when the value cannot be changed. The default for this field is the value specified for end positioning in the operational information format and can be changed to any of the following:

0	Rewind the tape volume.
1	Unload the tape volume.
2	Do not position tape (*LEAVE).
3	Unload and eject (remove) the tape volume.

File expiration date. The expiration date of the file being written in the header label to control data file expiration.

This field uses the Julian format with a leading century digit. It is of the form CYYDDD, where C represents the thousands and hundreds of the year value (19 is blank, 20 is 0, and 21 is 1), YY represents the year (00-99), and DDD represents the day of the year (001-366). For example, 1 February 1972 is represented as ' 72032' and 1 February 2072 is represented as '072032'.

The value *PERM indicates that the tape file is a permanent tape file. This special value must be in uppercase and left-justified. Blanks indicate that the default expiration date should be used.

You can change this value at SOF or SOV tape position exit type when the tape operation field is 1 and the initialize new volume label field is 0. (These fields are in the operational information format.) The default for this field is the file expiration date that the user requested.

If the field is changed to contain a date that is not valid, it is ignored and message CPF4063 is issued.

≫ Image catalog name. The name of the currently loaded image catalog on the virtual device. This field can be changed at any exit type along with setting the volume acceptance field to a value of 4 to cause a replacement virtual tape image catalog to be loaded. If the tape position exit type is SOF, SOV, or SOS, this field identifies a replacement for the current image catalog. If the tape position exit type is EOS or EOF, this field identifies a replacement for the next image catalog.

This field is always set to blank when the device is not a virtual device. If an incorrect or unusable image catalog name is provided, or the device is not a virtual device, message CPF41B0 will be issued.

The field is ignored during Duplicate Tape (DUPTAP) operations on all tape position exit types except SOF, SOV, and SOS on the first source and first target volumes. ≪

Issue active file messages. Whether i5/OS should issue active file messages. If the field contains a value that is not valid, it is ignored and message CPF4067 is issued. You can change the value when the tape position exit type field is SOV. The default is 0 when the tape position exit type is SOV. Valid values are:

0	Issue active file messages
1	Do not issue active file messages

Issue mount next tape message. Whether i5/OS should issue mount next tape messages. If the field contains a value that is not valid, it is ignored and message CPF4067 is issued. You can change this value when the tape position exit type field is EOS. The default is 0 when the tape position exit type is EOS.

0	Mount next tape messages are issued.

1   Mount next tape messages are not issued.

Large sequence number. The sequence number being used. At the start-of-volume exit type, this field contains the user-specified sequence number. This field is blank for all other exit types. The field can be changed for both input and output operations when the sequence number change allowed field in the operational information indicates that a change is allowed.

For output operations, the following user-specified sequence numbers may be passed to the tape management system:
- blank
- *END (left-justified and padded with blanks)
- A numeric value (right-justified with leading zeros or blanks) from 1 to 16 777 215

The sequence number may be changed to any of the following values:
- blank
- *END (left-justified and padded with blanks)
- A numeric value (right-justified with leading zeros or blanks) from 1 to 16 777 215

For input operations, the following user-specified sequence numbers may be passed to the tape management system:
- blank
- *FIRST (left-justified and padded with blanks)
- *NEXT (left-justified and padded with blanks)
- A numeric value (right-justified with leading zeros or blanks) from 1 to 16 777 215

The sequence number may be changed to any of the following values:
- blank
- A numeric value (right-justified with leading zeros or blanks) from 1 to 16 777 215

An invalid value or changing the sequence number when it is not allowed causes message CPF416B to be issued. This field overrides a value that is specified for the sequence number field. This field has the same function as the sequence number field, but allows for a greater range of sequence numbers.

Logical block identifier. The tape position to locate. This field is checked after the SOV tape position exit type is reached. If the logical block identifier is changed by the user at a tape position exit type other than SOV, or is changed to an incorrect value, the logical block identifier is referred to as an incorrect logical block identifier in message CPD4076. The logical block identifier is an incorrect value for any of the following:
- The tape is the wrong format
- The tape is not a standard label (*SL) tape
- The tape is not opened for input
- The tape is opened for bypass label processing (*BLP)
- The tape is opened for a read-backward operation
- The device does not support positioning by logical block identifier
- The identifier is not found on the tape

If the value specified is not a valid logical block ID, the value is ignored and message CPD4076 is issued. If the value is ignored, the tape positioning is done by sequence numbers. You can change this value at the SOV tape position exit type. The default is 0 at the SOV tape position exit type.

Message response. The response to the message exit type. You can change the value when the tape processing exit type is message. The default is 0 when the tape processing exit point is message. If the message response from the user is anything other than 0, a message that states that the tape management system handled the error is sent to the user. Valid values are as follows:

0	Ignore the message exit type. The message is sent when i5/OS regains control of the program from the user exit program.
1	Cancel the operation by replying to the message exit type with a C before the message is actually sent. This stops the message from being sent.
2	Ignore the operation by replying to the message exit type with an I before the message is actually sent. This stops the message from being sent.
3	Retry the operation by replying to the message exit type with an R before the message is actually sent. This stops the message from being sent.
4	Initialize the tape volume by replying to the message exit type with an INZ before the message is actually sent. This stops the message from being sent.
5	Continue the operation by replying to the message exit type with a G before the message is actually sent. This stops the message from being sent.

Mismatch acceptance. The control value that determines how to handle a cartridge ID mismatch situation. You can change the value when the tape library device exit type field is mismatch. The default is 1 when the tape library device exit type is mismatch. Valid values are:

1	Ignore the mismatch of the cartridge identifier and the logical volume identifier.
2	Initialize the logical volume identifier to match the cartridge identifier.
3	Eject the tape cartridge immediately from the tape library device and place it in the *EJECT category. The tape operation ends immediately.
4	Reject the output operation, leaving the tape cartridge in the tape library device for input operations. The tape operation ends immediately.

≫ Suppress auto-generate of virtual tape volume. Whether to suppress auto-generate of a new virtual tape volume. You can change this value when the tape position exit type field is EOS. The default is 0 when the tape position exit type is EOS. Valid values are:

0	Allow a new virtual tape volume to be auto-generated when the end of the image catalog is reached.
1	Do not allow a new virtual tape volume to be auto-generated when the end of the image catalog is reached. ≪

Sequence number. The sequence number to be used. Use the large sequence number field rather than this field if you are adding support for this function to a tape management system. At the start-of-volume exit type, this field contains the user-specified sequence number. This field is blank for all other exit types. The field can be changed for both input and output operations when the sequence number change allowed field in the operational information indicates that a change is allowed.

For output operations, the following user-specified sequence numbers may be passed to the tape management system:

- blank
- *END (left-justified and padded with blanks)
- A numeric value (right-justified with leading zeros or blanks) from 1 to 999 999

The sequence number may be changed to any of the following values:

- blank
- *END (left-justified and padded with blanks)
- A numeric value (right-justified with leading zeros or blanks) from 1 to 999 999

For input operations, the following user-specified sequence numbers may be passed to the tape management system:

- blank
- *FIRST (left-justified and padded with blanks)
- *NEXT (left-justified and padded with blanks)
- A numeric value (right-justified with leading zeros or blanks) from 1 to 999 999

The sequence number may be changed to any of the following values:

- blank
- A numeric value (right-justified with leading zeros or blanks) from 1 to 999 999

An invalid value or changing the sequence number when it is not allowed causes message CPF416B to be issued. This field is ignored if a value is specified for the large sequence number field.

Use optimum block size. Whether to use the optimum block size for save commands. If the optimum block size is used on a save command, commands such as Duplicate Tape (DUPTAP) only duplicate the tape volume to devices that support the same block size. If the optimum block size is not used, normal save processing uses the default block size that can be duplicated to any device type by using the DUPTAP command. The field is initialized to the value that the user specifies on the save command. The field can be changed at the start-of-command exit type only. At all other exit points, the field is set to blanks. Note that if a blank is passed, the value cannot be overridden by the tape management system. For example, on the Save System (SAVSYS) command, the value specified by the user for Use Optimum Block (USEOPTBLK) cannot be changed. Valid values follow:

0	Do not use optimum block size
1	Use optimum block size
blank	An override is not allowed for optimum block size

Volume acceptance. Whether the exit program accepts the mounted volume ≫ or the loaded virtual tape image catalog. ≪ This field has precedence over all other fields. You can change this value when the tape position exit type field is 1 through 6. The default is 1 for tape position exit types 1 through 5.

1	Mounted volume is accepted
2	No volume is accepted. The tape operation ends immediately.
3	Reject in favor of another volume
4	Reject and unload in favor of another volume ≫ or in favor of another virtual tape image catalog. When a new virtual tape image catalog is specified a new virtual tape volume must also be provided. ≪

An acceptance value of '3' is not allowed when a category is mounted in a tape library device. CPF410B, CPF450B, or CPF510B will be issued and the tape operation ends immediately. An acceptance value of '4' is not allowed for a tape position exit type of SOV when the Initialize new volume label field of the Operational Information is set to '1'. ≫ An acceptance value of '3' is not allowed when rejecting in favor of another virtual tape image catalog. ≪

Volume to be used. The volume to be loaded if 3 or 4 is specified for the volume acceptance field. This field can be set at any exit type to pass a replacement volume identifier. If the tape position exit type is SOF, SOV, or SOS, this field identifies a replacement for the current volume identifier. If the tape position exit type is EOS or EOF, this field identifies a replacement for the next volume identifier.

The field is ignored during Duplicate Tape (DUPTAP) operations on all tape position exit types except SOF, SOV, and SOS on the first source and first target volumes.

The field is used as a new volume identifier override during Initialize Tape (INZTAP) operations.

The field must be set to blanks if 4 is specified for the volume acceptance field when a category is mounted in a tape library device.

Error Messages

Message ID	Error Message Text
CPD4003 D	Alternate volume identifier not standard.
CPD4076 D	Logical block identifier not correct.
CPF4062 D	Alternate volume identifier not standard.
CPF4063 D	Incorrect file expiration date &7 specified.
CPF4067 D	Incorrect user exit control value specified.
CPF410A E	Error detected using program &6 in &7.
CPF410B E	Volume acceptance response not valid.
CPF410C E	End request specified by program &6.
CPF416B E	Incorrect user exit control value specified.
≫ CPF41B0 E	Incorrect image catalog name specified. ≪
CPF4402 D	Alternate volume identifier not standard.
CPF450A E	Error detected using program &6 in &7.
CPF450B E	Volume acceptance response not valid.
CPF450C E	End request specified by program &6.
CPF510A E	Error detected using program &6 in &7.
CPF510B E	Volume acceptance response not valid.
CPF510C E	End request specified by program &6.
CPF5401 E	Interface error on device &4.

Exit program introduced: V3R1

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

```
IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY  10504-1785
U.S.A.
```

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

```
IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan
```

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.
```

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) IBM 2006. Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. 1998, 2006. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This Application Programming Interfaces (API) publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced 36
Advanced Function Printing
Advanced Peer-to-Peer Networking
AFP
AIX
AS/400
COBOL/400
CUA
DB2
DB2 Universal Database
Distributed Relational Database Architecture
Domino
DPI
DRDA
eServer
GDDM
IBM
Integrated Language Environment
Intelligent Printer Data Stream
IPDS
i5/OS
iSeries
Lotus Notes
MVS
Netfinity
Net.Data
NetView
Notes
OfficeVision
Operating System/2
Operating System/400
OS/2
OS/400
PartnerWorld
PowerPC
PrintManager
Print Services Facility
RISC System/6000
RPG/400
RS/6000
SAA
SecureWay
System/36
System/370
System/38
System/390
VisualAge
WebSphere
xSeries

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and Conditions

Permissions for the use of these Publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these Publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these Publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE

IBM ®

Printed in USA