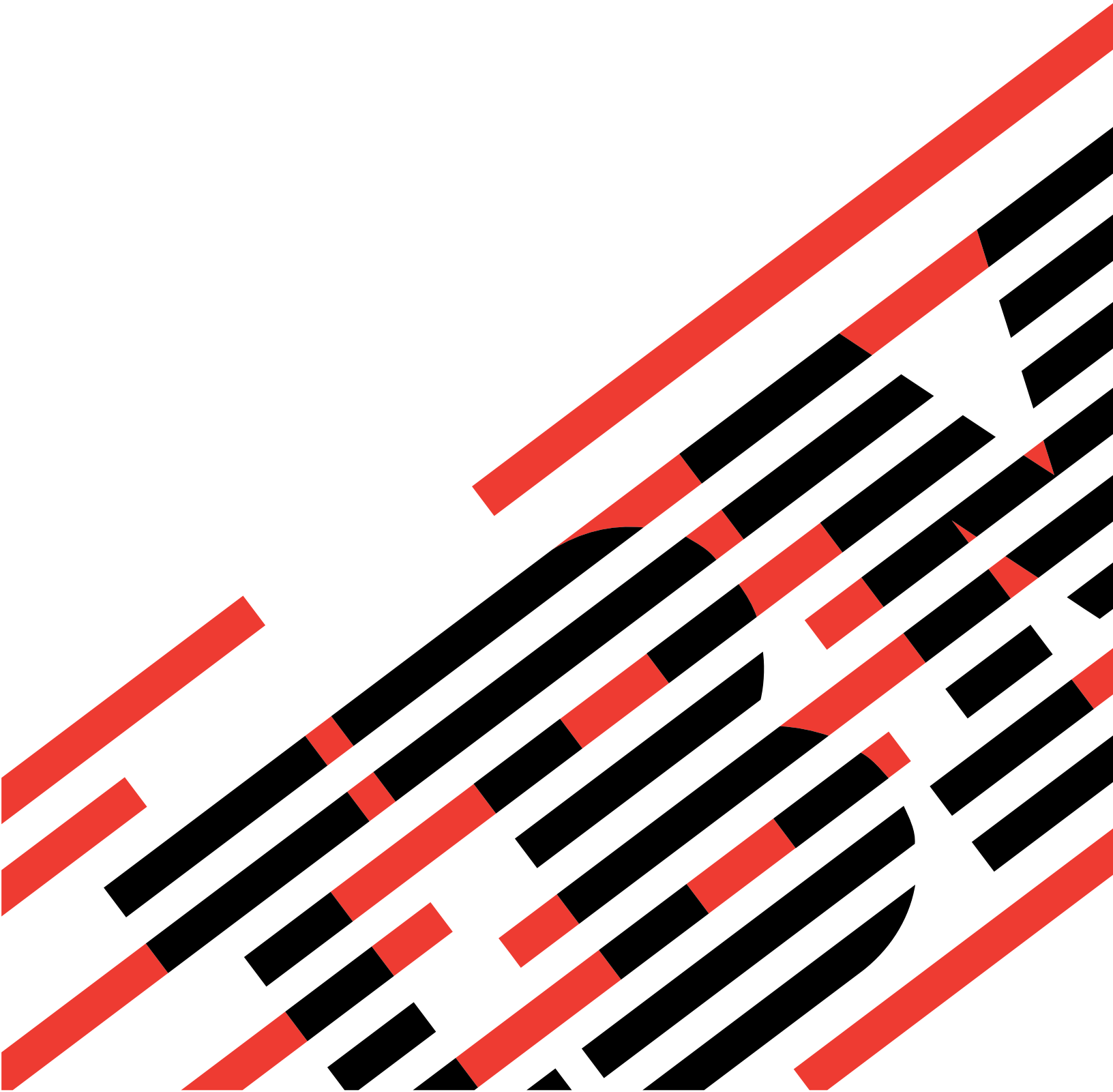
IBM

eserver

iSeries

CL Commands Volume 9



IBM

@server

iSeries

CL Commands Volume 9

Contents

Command Descriptions

CRTQMFORM (Create Query Management Form) Command Description

CRTQMFORM Command syntax diagram

Purpose

The Create Query Management Form (CRTQMFORM) command allows the user to create a query management form from a specified source file member. The query management form defines how a report is to look when data from running a query is displayed or printed.

Required Parameter

QMFORM

Specifies the name of the query management form to be created.

The possible library values are:

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

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

form-name: Specify the name of the form to be created.

Optional Parameters

SRCFILE

Specifies the qualified name for the source file containing the source for the form being created. The form is identified by the name specified on the QMFORM parameter.

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

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

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

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

QQMFORMSRC: The file having the IBM-supplied source file name, QQMFORMSRC, contains the source for the form to be created.

source-file-name: Specify the name of the file containing the source for the form to be created.

SRCMBR

Specifies the name of the source file member containing the form source to be created.

***QMFORM:** The member name is the same as the form name specified on the QMFORM parameter.

source-file-member-name: Specify the name of the member that contains the form source.

TEXT Specifies the text that briefly describes the query management form. More information on this parameter is in Commonly used parameters.

***SRCMBRTXT:** The descriptive text for the object being created is the same as the text description of the source member. If the source member has no text description, or if it is blank, then TEXT(*BLANK) is assumed.

***BLANK:** No text is specified. If this command causes an object to be replaced, then any existing text for the object is replaced with blank text.

'description': Specify no more than 50 characters of text, enclosed in apostrophes. The apostrophes are not part of the 50-character string.

AUT Specifies the authority given to users who do not have specific authority to the query management form, who are not on an authorization list, and whose user group has no specific authority to the query management form. More information on this parameter is in Commonly used parameters.

***LIBCRTAUT:** The public authority for the query management form is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the query management form). The public authority is determined when the query management form is created. If the CRTAUT value for the library changes after the query management form is created, the new value does not affect any existing objects.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the query management form. If the object is part of an authorization list, the user cannot add, change, or remove other users.

***USE:** The user can perform basic operations on the object, for example, run a program or read a file. The user is prevented from changing the object in any way. This authority provides object operational authority and read authority.

***EXCLUDE:** The user cannot access an object.

authorization-list-name: Specify the name of the authorization list used.

REPLACE

Specifies whether an existing object with the same name and type in the library to receive the output is replaced with the output of this command. More information on this parameter is in Commonly used parameters.

***YES:** An existing object is replaced with the output of this command.

***NO:** An existing object is not replaced with the output of this command.

Example for CRTQMFORM

```
CRTQMFORM   QMFORM(FORMEMP)  SRCFILE(RPTLIB/FORMSRC)
```

This command creates a form named FORMEMP in the current library. The form source is in member FORMEMP, which is located in source file FORMSRC in library RPTLIB.

Error messages for CRTQMFORM

***ESCAPE Messages**

QWM2701
&1 command failed.

QWM2703
&1 command ended.

QWM2705
Source file &1 in &2 not available.

QWM2706
&1 in &2 not replaced.

CRTQMQRY (Create Query Management Query) Command Description

CRTQMQRY Command syntax diagram

Purpose

The Create Query Management Query (CRTQMQRY) command allows the user to create a query management query from a specified source file member. The source for a query is a single Structured Query Language (SQL) statement that can contain variable substitution values and embedded comments. It can be spread over multiple records in a source file member.

Required Parameter

QMQRY
Specifies the name of the query management query to be created.

The possible library values are:

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

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

query-name: Specify the name of the query to be created.

Optional Parameters

SRCFILE
Specifies the qualified name for the source file containing the source for the query being created. The query is identified by the name specified on the QMQRY parameter.

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

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

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

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

QQMQRYSRC: The file having the IBM-supplied source file name, QQMQRYSRC, contains the source for the query to be created.

source-file-name: Specify the name of the file containing the source for the query to be created.

SRCMBR
Specifies the name of the source file member containing the source for the query to be created.

***QMQRY:** The member name is the same as the query name specified on the QMQRY parameter.

source-file-member-name: Specify the name of the member that contains the query source.

TEXT Specifies the text that briefly describes the query management query. More information on this parameter is in Commonly used parameters.

***SRCMBRTXT:** The descriptive text for the object being created is the same as the text description of the source member. If the source member has no text description, or if it is blank, then TEXT(*BLANK) is assumed.

***BLANK:** No text is specified. If this command causes an object to be replaced, then any existing text for the object is replaced with blank text.

'description': Specify no more than 50 characters of text, enclosed in apostrophes. The apostrophes are not part of the 50-character string.

AUT Specifies the authority given to users who do not have specific authority to the query management query, who are not on an authorization list, and whose user group has no specific authority to the query management query. More information on this parameter is in Commonly used parameters.

***LIBCRTAUT:** The public authority for the query management query is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the query management query). The public authority is determined when the query management query is created. If the CRTAUT value for the library changes after the query management query is created, the new value does not affect any existing objects.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the query management query. If the object is part of an authorization list, the user cannot add, change, or remove other users.

***USE:** The user can perform basic operations on the object, for example, run a program or read a file. The user is prevented from changing the object in any way. This authority provides object operational authority and read authority.

***EXCLUDE:** The user cannot access an object.

authorization-list-name: Specify the name of the authorization list used.

REPLACE
Specifies whether an existing object with the same name and type in the library to receive the output is replaced with the output of this command. More information on this parameter is in Commonly used parameters.

***YES:** An existing object is replaced with the output of this command.

***NO:** An existing object is not replaced with the output of this command.

SRTSEQ
Specifies the sort sequence table to be used for string comparisons in this query.

Note: This parameter is valid only when creating a query. This parameter is not valid when creating a FORM.

***SRC:** The source file member contains the sort sequence to be used when creating the query.

***JOBRUN:** The SRTSEQ value for the job at the time the query is run is used.

***JOB:** The SRTSEQ value for the job at the time the query is created is used.

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

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

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

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

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

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

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

table-name: Specify the name of the sort sequence table to be used when this query is created.

LANGID

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

***SRC:** The source file member contains the language ID to be used when the query is created.

***JOBRUN:** The LANGID value for the job is determined when the query is run.

***JOB:** The LANGID value for the job is determined when the query is created.

language-ID: Specify the language identifier to be used for the query.

Example for CRTQMQRY

```
CRTQMQRY   QMQRY(NEWQRY)  SRCFILE(RPTLIB/QRYSRC)
```

This command creates a query named NEWQRY in the current library. The query source is in member NEWQRY, which is located in source file QRYSRC in library RPTLIB.

Error messages for CRTQMQRY

*ESCAPE Messages

QWM2701

&1 command failed.

QWM2703
&1 command ended.

QWM2705
Source file &1 in &2 not available.

QWM2706
&1 in &2 not replaced.

CRTQSTDB (Create Question-and-Answer Database) Command Description

CRTQSTDB Command syntax diagram

Purpose

The Create Question-and-Answer Database (CRTQSTDB) command allows the user to create a new question-and-answer (Q & A) database. More information is available in the Basic System Operations topic in the Information Center.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The user of this command must have *ADD authority to the library in which the database files are located.

Optional Parameters

QSTDB
Specifies the Q & A database to create. When selecting a Q & A database name, the user must specify 10 or less characters and the name must begin with a letter.

***SELECT:** The user is asked to specify a Q & A* database. If only one Q & A database exists on the system, it is the default.

question-database: Specify the name of the Q & A database in which to create.

LIB Specifies the name of an existing library that will contain the new Q & A database.

QUSRSYS: The Q & A database is created in the QUSRSYS library.

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

Note: The library must exist on the system.

Example for CRTQSTDB

```
CRTQSTDB
```

This command shows the Create a Q & A Database display.

Error messages for CRTQSTDB

None

CRTQSTLOD (Create Question-and-Answer Load) Command Description

CRTQSTLOD Command syntax diagram

Purpose

The Create Question-and-Answer Load (CRTQSTLOD) command allows the user to create for distribution a Question-and-Answer (Q & A) database load on an alternative medium, such as tape. More information is available in the Basic System Operations topic in the Information Center.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The user must have *READ authority to the Q & A database file referred to by the command.

Optional Parameters

QSTDB

Specifies the Q & A database to distribute.

***SELECT:** The user is asked to specify a Q & A* database. If only one Q & A database exists on the system, it is the default.

question-database: Specify the name of the Q & A database that you want to distribute.

LIB Specifies the name of the library that contains the Q & A database to be used to create the load.

***QSTLIB:** The library containing the specified Q & A* database is searched. If *SELECT is specified on the QSTDB parameter, any Q & A database in any library for which the user is authorized can be selected.

library-name: Specify the name of the library to be searched. If *SELECT is specified on the QSTDB parameter, any database in the library for which the user is authorized can be selected.

Example for CRTQSTLOD

```
CRTQSTLOD
```

This command shows the Create a Database Load display.

Error messages for CRTQSTLOD

None

CRTSAVF (Create Save File) Command Description

CRTSAVF Command syntax diagram

Purpose

The Create Save File (CRTSAVF) command creates an online save file. An online save file is used with save and restore commands to retain data that would otherwise be written to tape or diskette. The file can also be used like a database file to read or write records that contain save/restore information. An online save file can also be used to send objects to another user on the SNADS network. For more information on sending an online save file, see the Send Network File (SNDNETF) command.

Restrictions:

1. An online save file should only be used for save/restore data. When restoring objects from the save file, the data in the file must have been produced by a save command. When records are written to a save file by using a high-level language program, the records must contain data produced by saving objects into the save file.
2. The records retrieved from a save file contain sequencing checksum information that is validated by the system when records are inserted into a save file. An attempt to insert records that are either out of sequence or have been changed since retrieval from a save file are rejected.
3. A save file can contain the output of only one save operation and one library, and all objects in the save file must have been saved from the same library. If the SNDNETF command is used to send a save file, the maximum size of the save file is approximately 2 billion bytes. The number of objects saved in a save file is identical to the number of objects saved when saving to tape or diskette, as long as the file's maximum size is not exceeded. The exact number of objects saved into a save file depends on the object types and the object contents.
4. The user must have object operational authority to the CRTSAVF command to create a duplicate save file or to restore a save file.
5. The user must have add and read authority to the library in which the save file is located.

Required Parameter

FILE Specifies the qualified name of the file being created. If the file is used by a high-level language (HLL) program, the file name must be consistent with the naming rules of that language; otherwise, the file must be renamed in the program.

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

***CURLIB:** The save file is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the save file is created.

file-name: Specify the name of the file being created.

Optional Parameters

MAXRCDS

Specifies the maximum number of records the save file can hold. The size of the save file is estimated in bytes about 8192 + (512 x number of records in the save file).

There is room for approximately two thousand 512-byte records in 1 megabyte of space. To ensure that the save file does not exceed approximately 20 megabytes (20 x 2,000), specify MAXRCDS (40000).

Note: The maximum amount of data that a save file can contain is approximately one terabyte (1,099,511,627,776 bytes).

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

number-of-records: Specify the maximum number of records that the save file can contain. Valid values range from 1 through 2146762800.

ASP Specifies the auxiliary storage pool (ASP) in which the system creates the save file.

***LIBASP:** The save file is created in the same ASP as the one containing the library holding the file.

ASP-identifier: Specify a value ranging from 1 through 16 for the ASP identifier. These values can only be used if the ASP has already been configured on the system.

Note: The value of 1 is the system ASP and is always configured on the system.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Commonly used parameters.

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds that the program waits for the file resources to be allocated. Valid values range from 1 through 32767 seconds.

SHARE

Specifies whether the open data path (ODP) for the save file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the Database Programming topic in the Information Center.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

This includes several opens in the same program.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

AUT Specifies the authority given to users who do not have specific authority to the save file, who are not on an authorization list, and whose user group has no specific authority to the save file. More information on this parameter is in Commonly used parameters.

***EXCLUDE:** The user cannot access the save file.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the save file.

***CHANGE:** The user can perform all operations on the save file except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the save file. Change authority provides object operational authority and all data authority.

***LIBCRTAUT:** The public authority for the save file is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the save file). The public authority is determined when the save file is created. If the CRTAUT value for the library changes after the save file is created, the new value does not affect any existing objects.

***USE:** The user can perform basic operations on the save file, such as running a program or reading a file. The user cannot change the save file. *USE authority provides object operational authority, read authority, and execute authority.

authorization-list-name: Specify the name of the authorization list used.

TEXT Specifies the text that briefly describes the save file. More information on this parameter is in Commonly used parameters.

***BLANK:** Text is not specified.

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

Example for CRTSAVF

```
CRTSAVF   FILE(ONLINE)  TEXT('Online save file')
```

This command creates an online save file named ONLINE in the current library. The save file is in the system ASP with no maximum number of records. The public has no authority to this file; only the object owner and users that have the object owner user profile as their group profile can use this save file.

Error messages for CRTSAVF

***ESCAPE Messages**

CPF7302
File &1 not created in library &2.

CRTSCHIDX (Create Search Index) Command Description

CRTSCHIDX Command syntax diagram

Purpose

The Create Search Index (CRTSCHIDX) command is used to create a search index. A search index is a reference to the help information contained in one or more panel groups.

The user can access a search index through data description specifications (DDS), by pressing the Help key, or through the information search function using the Start Search Index (STRSCHIDX) command.

A search index created by the CRTSCHIDX command does not contain any data. The user can add data by using the Add Search Index Entry (ADDSCHIDXE) command.

Restrictions:

1. The user of this command must have *ADD authority for the library where the search index is located.
2. IBM-supplied panel groups cannot be added to a search index created with the CRTSCHIDX command; only panel groups created with the Create Panel Group (CRTPNLGRP) command can be added to a search index created with the CRTSCHIDX command.

Required Parameters

SCHIDX

Specifies the qualified name of the search index being created.

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

***CURLIB:** The search index is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the search index is created.

search-index-name: Specify the name of the search index.

TITLE Specifies the title to appear at the top of each panel when the search information is presented.

'panel-title': Specify no more than 55 characters of text, enclosed in apostrophes.

Optional Parameters

CHRID

Specifies the graphic character set and code page values used for the search index. The value of TXTCHRID for panel groups added must match the value of this parameter. (TXTCHRID is an attribute on a UIM panel group tag.)

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

Element 1: Character-set

character-set: Specify the graphic character set values that match the code page of the synonyms used in the search index.

Element 2: Code-page

code-page: Specify the code page value that matches the code page of the synonyms used in the search index.

AUT Specifies the authority given to users who do not have specific authority to the object, who are not on the authorization list, or whose user group has no specific authority to the object.

***LIBCRTAUT:** The authority for the library in which the search index is being created is the authority for the search index.

***ALL:** The user can perform all operations on the search index except those limited to the owner. The user can create, delete, change, and specify security for the search index. The user can transfer ownership of the search index.

***USE:** The user can perform basic operations on the search index, such as running a program or reading a file. The user cannot change the search index. *USE authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The user cannot access the search index.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

authorization-list-name: Specify the name of an authorization list from which the authority to the object is taken.

TEXT Specifies the text that briefly describes the search index. More information on this parameter is in Commonly used parameters.

***TITLE:** The first 50 characters of the title are used.

***BLANK:** Text is not specified.

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

Example for CRTSCHIDX

```
CRTSCHIDX   SCHIDX(ACCOUNTING)
  TITLE('Accounting Help Index')
  TEXT('Accounting Help Index')
```

This command creates a search index named ACCOUNTING in the current library.

Error messages for CRTSCHIDX

***ESCAPE Messages**

CPF6E11
Search index &2 not created in library &3.

CRTSRVPGM (Create Service Program) Command Description

CRTSRVPGM Command syntax diagram

Purpose

The Create Service Program (CRTSRVPGM) command creates a bound service program from a set of modules and binding directories.

Restrictions:

1. You must have *READ and *ADD authority for the library where the service program is being created.
2. You must have *USE authority to the specified modules, service programs, and binding directories.

Required Parameter

SRVPGM
Specifies the qualified name of the service program object that is created.

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

***CURLIB:** The service program object is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the service program object is created.

service-program-name: Specify the name of the created service program.

Optional Parameters

MODULE
Specifies the list of modules that are copied and bound together to create the service program object. If duplicate module and library specifications are found, only the first instance of the duplicate module and library is used. Modules in this list are copied into the final service program object.

***SRVPGM:** The module and library name specified on the SRVPGM parameter is used. The name specified on the SRVPGM parameter is used as the module name.

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

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

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

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

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

***ALL:** Find all module objects in the specified library or libraries.

module-name: Specify the name of the module that is copied to create the service program object.

generic-module-name:* Specify all module objects starting with the characters preceding the * in the specified library or libraries.

EXPORT

Specifies the names of the data and procedures this service program exports.

***SRCFILE:** The source file and source member specify the data and procedures to export from the service program.

***ALL:** All data and procedures that are exported from the included modules are also exported from the service program.

SRCFILE

Specifies the source file containing the specifications for exporting data and procedures from this service program.

QSRVSRC: The name of the source file containing the specifications for exporting data and procedures.

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

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

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

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

source-file-name: Specify the name of the source file containing the specifications for exporting data and procedures.

SRCMBR

Specifies the name of the source file member containing the specifications for exporting data and procedures from this service program.

***SRVPGM:** The source file member name is the same name as the service program name specified for the SRVPGM parameter.

source-file-member-name: Specify the name of the member in the source file containing the specifications for exporting data and procedures.

BNDSRVPGM

Specifies the list of service program exports to examine at bind time to ensure they satisfy any module import requests. The service program exports are checked only if there are unresolved module import requests not satisfied by the set of module exports. Any service program specified on the BNDSRVPGM parameter that satisfies a module import request will be bound to the program being created. The service program name and the library specified on the BNDSRVPGM parameter are saved to be used at run time.

***NONE:** No service programs are provided for symbol resolution.

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

***LIBL:** All libraries in the job's library list are searched until the first match is found. If one of the service programs has an export that satisfies a module import, then the library list is searched at run-time to find this service program.

library-name: Specify the name of the library where the service programs can be found. If one of the service programs has an export that satisfies a module import, and a specific library was specified, the specified library is searched at run-time to find this service program.

Note: QTEMP is not a valid library name for this parameter.

***ALL:** Find all service program objects in the specified library or libraries.

Note: This value should only be specified in a user-controlled environment when you know exactly what is getting bound to your program. Specifying *LIBL with *ALL may give you unpredictable results at program run time. Specify the generic service program name or specific libraries to better control what gets bound to your program.

generic-service-program-name:* Specify a generic name of the service program. All service program objects starting with the characters preceding the * in the specified library or libraries are examined.

service-program-name: Specify the name of the service program to be examined during symbol resolution.

BNDDIR

Specifies the list of binding directories that are used in symbol resolution. The exports of the modules and service programs in the binding directory are only checked if there are unresolved module import requests that the exports from the modules and service programs (specified in the MODULE or BNDSRVPGM parameters) could not satisfy.

***NONE:** No binding directory is specified.

The name of the specified binding directory can be qualified by one of the following library values:

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

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

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

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

binding-directory-name: Specify the name of the binding directory used in symbol resolution.

ACTGRP

Specifies the activation group this service program is associated with when it is called. An activation group provides:

- Run-time data structures to support the running of programs
- Addressing protection
- A logical boundary for message propagation
- A logical boundary for application cleanup processing

***CALLER:** When this service program gets called, the program is activated into the caller's activation group.

activation-group-name: Specify the name of the group that is associated with this called service program. If an activation group, by the specified name, currently exists when this service program is called, the service program is associated with the already existing activation group. If an activation group by the specified name does not currently exist when this service program is called, then a new activation group is created and the service program is associated with the newly created activation group.

OPTION

Specifies options to be used when the service program object is created.

Service Program Creation Options

***GEN:** A service program object is generated.

***NOGEN:** A service program object is not generated.

Duplicate Procedure Name Options

***NODUPPROC:** During the symbol resolution phase of the binding process, each procedure name that is exported from the modules and service programs must be unique.

***DUPPROC:** During the symbol resolution phase of the binding process, the procedure names that are exported from the modules and service programs do not have to be unique. When multiple duplicate procedures are allowed, the first exported procedure in the list of specified modules and service programs that matches the import request is the procedure that is selected.

Duplicate Variable Name Options

***NODUPVAR:** During the symbol resolution phase of the binding process, each variable name that is exported from the modules and service programs must be unique.

***DUPVAR:** During the symbol resolution phase of the binding process, the variable names that are exported from the modules and service programs do not have to be unique. When multiple duplicate variables are allowed, the first exported variable in the list of specified modules and service programs that matches the import request is the variable that is selected.

Diagnostic Message Options

***WARN:** If duplicate variables or procedures are found, a diagnostic message is issued indicating what duplicates were found.

***NOWARN:** If duplicate variables and procedures are found, diagnostic messages are not issued.

Resolving References Options

***RSLVREF:** All imports must be resolved to exports for the service program to be created.

***UNRSLVREF:** All imports do not need to resolve to exports for the service program to be created. If the service program tries to use one of these unresolved imports at run time, an MCH3203 exception message will be issued.

DETAIL

Specifies the level of detail to be printed.

***NONE:** A listing is not generated.

***BASIC:** Contains a listing of the options passed to CRTPGM, and processing statistics. This listing also contains the Brief Summary Table.

***EXTENDED:** In addition to the information provided in the **BASIC* listing, this listing contains the Extended Summary Table and the Binding Information Listing.

***FULL:** This listing contains the **EXTENDED* listing and the Cross-Reference Listing.

Note: If a printed listing is requested, the print file *LIBL/QSYSPRT is used to generate the listing.

ALWUPD

Specifies whether to allow an update of the service program being created using the Update Service Program (UPDSRVPGM) command.

***YES:** The service program can be updated in the future using the UPDSRVPGM command.

***NO:** The UPDSRVPGM command does not update the service program being created.

ALWLIBUPD

Specify whether to allow the bound service program library name to be changed on update, using the UPDSRVPGM command, of the service program being created.

***NO:** The UPDSRVPGM command will never be allowed to update the bound service program library names of the service program being created, even if ALWUPD is *YES.

***YES:** This service program may have the bound service program library names updated at any future time using the UPDSRVPGM command, as long as the ALWUPD is also *YES.

USRPRF

Specifies whether authority checking is performed only for the user running the program, or for both the user running the program and the program owner.

***USER:** The user profile of the program user is used when the program is run.

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

REPLACE

Specifies whether the existing program is replaced if a program by the same name already exists in the specified library.

***YES:** Replace the existing program by moving it to the QRPLOBJ library.

Note: Both programs must be owned by the same user for the replace to work.

***NO:** No replacement occurs.

AUT Specifies the authority given to users who do not have specific authority to the service program, who are not on an authorization list, and whose user group has no specific authority to the service program.

***LIBCRTAUT:** The public authority for the service program is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the service program). The public authority is determined when the service program is created. If the CRTAUT value for the library changes after the service program is created, the new value does not affect any existing objects.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the service program.

***USE:** The user can perform basic operations on the service program, such as running a program or reading a file. The user cannot change the service program. *USE authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The user cannot access the service program.

authorization-list: The authorization list must exist when the object is created. Users are granted authority to the object as specified by the list.

TEXT Specifies the text that briefly describes the program object. More information on this parameter is in Commonly used parameters.

***BLANK:** Text is not specified.

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

TGTRLS

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

When specifying the *target-release* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V3R6M0 is version 3, release 6, modification level 0.

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

 book for a complete list of valid values.

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

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

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

ALWRINZ

Specifies if the static storage of the service program is allowed to be reinitialized while it is still active.

***NO:** The static storage of the service program can not be reinitialized while it is still active.

***YES:** The static storage of the service program is allowed to be reinitialized while the service program is still active.

STGMDL
Specifies the storage model attribute of the service program.

***SNGLVL:** The service program is created with single-level storage model. When a single-level storage model service program is activated and run, it is supplied single-level storage for automatic and static storage. A single-level storage service program runs only in a single-level storage activation group.

***TERASPACE:** The service program is created with teraspace storage model. When a teraspace storage model service program is activated and run, it is supplied teraspace storage for automatic and static storage. A teraspace storage service program runs only in a teraspace storage activation group.

***INHERIT:** The service program is created with inherit storage model. When activated, the service program adopts the storage model of the activation group into which it is activated. An equivalent view is that it inherits the storage model of its caller. When the *INHERIT storage model is selected, the *CALLER activation group (ACTGRP parameter) must also be selected.

IPA Specifies whether interprocedural analysis (IPA) is to be used during the service program creation.

For more information on IPA, refer to the ILE Concepts  book.

***NO:** Interprocedural analysis will not be performed.

***YES:** Interprocedural analysis will be performed. Note that the resulting service program will not be allowed to be updated with the Update Service Program (UPDSRVPGM) command.

IPACTLFILE
Gives the path name of a file which contains interprocedural analysis (IPA) suboption information. This parameter is allowed only when IPA(*YES) is specified.

***NONE:** No IPA control file information is to be used when IPA(*YES) is specified.

'IPA-control-file': Specifies the path name of the IPA control file to use when IPA(*YES) is specified. If the name is qualified it must be enclosed in apostrophes. An example of a qualified IPA control file name is '/directory1/directory2/myipactlfname'

Example for CRTSRVPGM

```
CRTSRVPGM   SRVPGM(WORKDOC)
```

This command creates the service program object named WORKDOC in the current library.

Error messages for CRTSRVPGM

*ESCAPE Messages

CPF223E
Authority check for use adopted authority attribute failed.

CPF5D05
Service program &1 not created.

CPF5D07
Export source file record length greater than 240.

CPF5D12
Error encountered during program or service program preparation.

CRTSRCPF (Create Source Physical File) Command Description

CRTSRCPF Command syntax diagram

Purpose

The Create Source Physical File (CRTSRCPF) command creates a source physical file in the database. A source physical file is created from the file description parameters in the CRTSRCPF command; it is used to store source records that are used as input to IBM-supplied source processors, such as the data description specifications (DDS) processor, CL compiler, or RPG III compiler. To override attributes of the file after it has been created, use the Override Database File (OVRDBF) command before the file is opened. To change attributes of the file after it has been created, use the Change Source Physical File (CHGSRCPF) command.

Restrictions

1. This command is not threadsafe for DDM files of type *SNA, when SYSTEM(*RMT) or SYSTEM(*FILETYPE) is specified.

Required Parameter

FILE Specifies the qualified name of the file being created. If the file is used by a high-level language (HLL) program, the file name must be consistent with the naming rules of that language; otherwise, the file must be renamed in the program.

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

***CURLIB:** The source physical file is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the source physical file is created.

source-physical-file-name: Specify the name of the source physical file being created.

Note: If a DDM file is specified, the source physical file (specified in the RMTFILE parameter of the CRTDDMF command) is created on a remote system (specified in the RMTLOCNAME parameter of the CRTDDMF command). See the SYSTEM parameter of this command.

Optional Parameters

RCDLEN
Specifies the length (in bytes) of the records being stored in the source file. The format of each record contains three fields: the sequence number of the record, a date field, and the source statement. The record format name is the same as the file itself, specified in the FILE parameter. More information on this parameter is in Commonly used parameters.

The RCDLEN parameter must provide six positions for the source sequence number, six positions for the date field, and at least one position for source start. If the Copy File (CPYF) command is used to copy records into the file, and the records are longer than the length specified, the records are truncated on the right. These fields are defined with fixed attributes and names, and have a keyed access path over the sequence number if ACCPTH(*KEYED) is specified.

92: The default record length is 92 characters. Six characters are for the record sequence number, 6 are for the record date, and the remaining 80 characters are for the source statement.

record-length: Specify the record length of each source record in the file. Valid values range from 13 through 32766.

Double-Byte Character Set Considerations:

If IGCDTA(*YES) is specified, the RCDLEN parameter must provide six positions for the source sequence number, six positions for the date field, and at least four positions for source start. Valid values range from 16 through 32766.

MBR Specifies the name of the source file member being added when the source file is created. Other members can be added to the file after it is created by using the ADDPFM command.

***NONE:** No member is added when the file is created.

***FILE:** The member being added has the same name as that of the source file that contains the member (specified in the FILE parameter).

source-file-member-name: Specify the name of the member that is added when the source file is created.

SYSTEM

Specifies the system on which the source physical file is created.

***LCL:** The source physical file is created on the local system.

***RMT:** The source physical file is created on a remote system that is using DDM. The file name specified on the FILE parameter must be the name of a DDM file (created by using the CRTDDMF command). The DDM file contains the name of the source physical file being created (RMTFILE parameter on the CRTDDMF command) on the remote system (RMTLOCNAME parameter on the CRTDDMF command).

***FILETYPE:** If the name specified on the FILE parameter is a DDM file, the physical file is created on the remote system specified by the RMTLOCNAME parameter on the CRTDDMF command for that DDM file. Otherwise, the name on the FILE parameter cannot be the name of an existing file, since a physical file of that name is created on the local system.

EXPDATE

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

Note: Attempting to open a member for which the expiration date has been exceeded sends an error message to the user.

***NONE:** No expiration date is specified.

expiration-date: Specify the date after which the source file member cannot be used. The date must be in the format specified by the QDATFMT and QDATSEP job attributes. The date must be enclosed in apostrophes if special characters are used in the format.

MAXMBRS

Specifies the maximum number of members that the source file being created can contain at any one time.

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

maximum-members: Specify the maximum number of members that the source file can contain. Valid values range from 1 through 32767 members per file.

ACCPTH

Specifies the type of access path used by all members of the file.

***ARRIVAL:** The access path governs the arrival sequence order. Using this parameter value reduces the size of the file and avoids maintenance of the keyed access path.

***KEYED:** The access path is of keyed sequence order.

More information on keyed access paths and the arrival sequence orders of source file access paths is in the Database Programming topic in the Information Center.

ACCPTHSIZ

Specifies the maximum size of auxiliary storage that can be occupied by access paths that are associated with keyed source physical files. This parameter does not apply to access paths that are created for logical files or for queries that refer to the data in a source physical file.

***MAX1TB:** The access paths associated with this file can occupy a maximum of one terabyte (1,099,511,627,776 bytes) of auxiliary storage.

Note: This value is not supported on releases of the system earlier than Version 3 Release 6 Modification 0 (V3R6M0). Therefore, if an attempt is made to save a physical file that has this attribute, and the save operation specifies a target release earlier than V3R6M0, the save operation might be unsuccessful, or if successful, the access paths are not saved. If the save operation is successful and the saved version of the file is then used to restore the physical file, the system rebuilds all of the access paths.

***MAX4GB:** The access paths associated with this file can occupy a maximum of four gigabytes (4,294,966,272 bytes) of auxiliary storage. This value provides compatibility with releases of the operating system earlier than Version 3 Release 6 Modification 0.

MAINT

Specifies the type of access path maintenance used for all members of the source file that have keyed access paths.

***IMMED:** The access path is maintained for each physical file member whether the source physical file is opened or closed. The access path is changed whenever a record is updated, added to, or deleted from a member of this file or a logical file member based on a member of this file.

***REBLD:** The access path is completely rebuilt when a file member is opened during program running. The access path is continuously maintained until the member is closed, then the access path maintenance ends.

***DLY:** The maintenance of the access path is delayed until the physical file member is opened for use. The access path is changed only for records that have been added, deleted, or changed since the file was last opened. While the file is open, all changes made to its members are immediately reflected in the access path of those members, no matter what is specified for the MAINT parameter. To prevent a lengthy rebuild time when the file is opened, *DLY should be specified only when the number of changes to the access path is small.

If the number of changes between a close and the next open reaches approximately 10 percent of the access path size, the system stops saving changes and the access path is completely rebuilt the next time the file is opened.

RECOVER

Specifies, for files having immediate or delayed maintenance on their access paths, when recovery processing of the file is performed after a system failure has occurred while the access path is being changed. This parameter is valid only for files with a keyed access path.

If *IMMED is specified for the MAINT parameter, the access path can be rebuilt during initial program load (IPL) (before any user can run a job), after IPL has ended (when jobs are running at the same time), or when the file is reopened. While the access path is being rebuilt, the file cannot be used by any job.

During the IPL, an Override Access Path Recovery display lists access paths that must be recovered and what the RECOVER parameter value is for each. The RECOVER parameter values on this display can be overridden. More information on this is in the Backup, Recovery, and Availability topic in the Information Center.

If *REBLD is specified for the MAINT parameter, the access path can be rebuilt the next time its file is opened.

***NO:** The access path of the file is not rebuilt during *IPL or *AFTIPL. The file's access path, if not valid, is rebuilt when the file is reopened.

***AFTIPL:** The file has its access path rebuilt after the IPL has been completed. This option allows other jobs not using this file to start processing immediately after the IPL has been completed. If a job tries to allocate the file while its access path is being rebuilt, a file- open exception occurs.

***IPL:** The file has its access path rebuilt during the IPL. This ensures that the file's access path is rebuilt before the first user program tries to use it. However, no jobs can start running until after all files that specify RECOVER(*IPL) have their access paths rebuilt.

FRCACCPTH

Specifies, for files with keyed access paths only, whether access path changes are forced to auxiliary storage along with the associated records in the file whenever the access path is changed. FRCACCPTH(*YES) minimizes (but does not remove) the possibility that an abnormal job end can cause damage to the access path, which requires it to be rebuilt.

***NO:** The access path and updated records are not forced to auxiliary storage whenever the access path is changed.

***YES:** The access path and updated records are forced to auxiliary storage whenever the access path is changed. If FRCACCPTH(*YES) is specified, MAINT(*REBLD) cannot be specified.

FRCACCPTH(*YES) slows the response time of the system if the access path is changed in an interactive job. If the access path is changed frequently, the overall performance of the system is somewhat decreased.

SIZE Specifies the *initial* number of records in each member of the file, the number of records in each increment that can be automatically added to the member size, and the number of times the increment is automatically applied. The number of records for each file member is specified as the number of records that can be placed in it. This includes deleted records.

When the maximum number of records has been reached, a message, stating that the member is full is sent to the system operator, offering the choice of ending the request or extending the member size. The operator can extend the member by 10% or by the number of records specified as the increment value, whichever is greater, each time the message is received.

A list of 3 values can be specified to indicate the initial size of each member and the automatic extensions that can be added when needed, or *NOMAX can be specified. If SIZE is not specified, SIZE(10000 1000 499) is assumed by the system.

Element 1: Number of Records

One of the following is used to specify the *initial* number of records in the member before an automatic extension of the member occurs. The ALLOCATE parameter determines when the required space for the initial number of records is allocated. If *YES is specified, the space is allocated when a new member is added. If *NO is specified, the initial space is allocated as determined by the system.

10000: Initially, up to 10,000 records can be inserted into each member of the file before an extension occurs.

number-of-records: Specify the number of records (ranging from 1 through 16777215) that can be inserted before an automatic extension occurs. If automatic extensions are not wanted, enter zeros for the second and third values in the list.

Element 2: Increment Value

One of the following is used to specify the number of records that can be additionally inserted in the member when the initial member size is exceeded and an automatic extension occurs. The minimum size of an increment is 10% of the size of the member at the time the maximum number of records is reached.

1000: The member size is increased by 10% or 1,000 records, whichever is greater.

increment-value: Specify the number of additional records (ranging from 0 through 32767) which, if greater than 10% of the size of the member when the maximum number of records is reached, are to be added to the member during an automatic extension.

If the number specified is not greater than 10% of the member size and not equal to zero, the member size is increased by 10%.

Specify 0 to prevent automatic extensions. This value must be 0 if the value for the number of increments is 0.

Element 3: Maximum Number of Increments

One of the following values is used to specify the maximum number of increments that can be automatically added to a file member. If 0 is specified for the increment amount, the number of increments need not be specified; 0 will be the default value instead of 499 and a file-member-full message is sent to the user issuing the command.

499: Up to 499 increments can be automatically added to the member size.

number-of-increments: Specify the maximum number of increments, ranging from 0 through 32767, that can be automatically added to the member. To prevent automatic extensions, specify a value of 0.

Other Single Values

If SIZE(*NOMAX) is specified, ALLOCATE(*NO) must also be specified.

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

ALLOCATE

Specifies whether storage space is allocated for the initial number of records (SIZE parameter) for each source file member as it is added. The allocation provides enough space to hold the initial number of records specified by the SIZE parameter. Later allocations, which occur when a record cannot be added to a member without exceeding its capacity, are determined by the system and by the SIZE parameter values.

***NO:** Minimum storage space is initially allocated when the member is added. The system determines when space allocations are necessary and determines the size of each allocation.

***YES:** Storage space is initially allocated as each member is added. The amount specified in the first value of the SIZE parameter (the number of records) is allocated. If the space cannot be

allocated, a message is sent to the user and the affected member is not added. If ALLOCATE(*YES) is specified, SIZE(*NOMAX) cannot be specified.

CONTIG

Specifies whether records in the initial allocation in each source file member are stored contiguously (next to each other) on auxiliary storage. If so, and the necessary contiguous space is not available, the system sends a message to the job log and allocates the storage space separately. The file is still usable. This parameter does not affect additional allocations needed later, which most likely would be noncontiguous.

***NO:** The storage space for each member does not have to be together.

***YES:** The system allocates contiguous space for the added members of the source file. If the members are stored separately, the user is notified and a message is put in the job log. If CONTIG(*YES) is specified, ALLOCATE(*YES) must also be specified.

UNIT This parameter is no longer supported. It exists solely for compatibility with releases earlier than Version 3 Release 6 Modification 0 of the AS/400 system. For information on using auxiliary storage pools (ASPs), refer to the Backup, Recovery, and Availability topic in the Information Center.

You can specify the value *ANY or a value ranging from 1 through 255 on this parameter. Specifies whether a file is stored on a specific auxiliary storage unit. The system attempts to allocate the storage space for the file and for all of its members and their associated access paths on the specified unit. This includes the initial allocation when each member is added and any extensions that occur later for each member in the file. If the system cannot allocate the storage space for each member on the specified unit, it allocates the space on any available unit and sends a message to the job log. The file is entirely usable in all cases.

The unit identifier is a number ranging from 1 through 255 that is assigned when a new disk device is configured. The user can display and change the configured disk device using the Work with Disk Devices display from the Start System Service Tool (STRSST) command. More information on the System Service Tool (SST) is in the Backup, Recovery, and Availability topic in the Information Center.

***ANY:** The storage space for the file and its members is allocated on any available auxiliary storage unit.

unit-identifier: Specify the storage unit where the system attempts to allocate storage space for the file and all of its members.

If the unit specified is part of any user auxiliary storage pool (auxiliary storage pools ranging from 2 through 16), the system allocates space from ASP 1, the system auxiliary storage pool.

FRCRATIO

Specifies the number of inserted, updated, or deleted records processed before being forced to auxiliary (permanent) storage. More information on this parameter is in Commonly used parameters.

If this physical file is being journaled, a larger force ratio should be specified. More information on journal management is in the Journal management article in the Information Center.

***NONE:** There is no forced write ratio; the system determines when the records are written to auxiliary storage.

number-of-records-before-force: Specify the number of inserted, updated, or deleted records processed before they are explicitly forced to auxiliary storage.

IGCDTA

Specifies whether the file contains double-byte character set (DBCS) data.

***NO:** The file does not process DBCS data.

***YES:** The file processes DBCS data.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Commonly used parameters.

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds a program waits for the file resources to be allocated to the job. Valid values range from 1 through 32767 seconds.

WAITRCD

Specifies the number of seconds that a program waits for a record to be updated or deleted, or for a record read in the commitment control environment with LCKLVL(*ALL) specified. More information on record locking is in the Database Programming topic in the Information Center. If the record is not allocated in the specified wait time, an error message is sent to the program.

60: A program waits 60 seconds for a requested record.

***IMMED:** The program does not wait; when a record is locked, an immediate allocation of the record is required.

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

number-of-seconds: Specify the number of seconds that a program waits for a record requested for reading, updating, or deleting. Valid values range from 1 through 32767 seconds.

CCSID

Specifies the coded character set identifier (CCSID) being used to describe character data in the fields of the source file.

A CCSID is a 16-bit number identifying a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and additional coding-related information that uniquely identifies the coded graphic representation used.

***JOB:** The current job's default CCSID is used.

***HEX:** The CCSID 66535 is used, which indicates that the character data in the fields is treated as bit data and is not converted.

coded-character-set-identifier: Specify the CCSID being used. More information on valid CCSIDs is in the Globalization topic in Information Center.

SHARE

Specifies whether the open data path (ODP) for the source physical file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the Database Programming topic in the Information Center.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file, provided the scope specified on the OPNSCOPE keyword for the subsequent open of the file is compatible with the scope of the original open.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

DLTPCT

Specifies the maximum percentage of deleted records allowed in any member in the source file. The percentage is based on the number of deleted records compared with the total record count in a member. The percentage check is made when any member of the file is closed or any logical file member based on any member of the file is closed. If the number of deleted records exceeds the percentage, a message is sent to the system history log (QHST) to inform the user.

***NONE:** No percentage is specified; the number of deleted records in the file members is not checked when a member is closed.

deleted-records-threshold-percentage: Specify, from 1 through 100, the largest percentage of deleted records allowed in any member in the file. If this percentage is exceeded, a message is sent to the system history log (QHST) when the file is closed.

ALWUPD

Specifies whether records are updated in the physical file. Records in a logical file can be updated only when the records in each physical file on which the logical file is based can be updated.

***YES:** Records can be updated in the physical file.

***NO:** Records cannot be updated in this physical file or in any logical file built over this physical file.

ALWDLT

Specifies whether records can be deleted from the physical file. Records in a logical file can be deleted only when the records in each physical file on which the logical file is based can be deleted.

***YES:** Records can be deleted in this physical file.

***NO:** Records cannot be deleted in this physical file or from any logical file built over this physical file.

AUT Specifies the authority given to users who do not have specific authority to the source physical file, who are not on an authorization list, and whose user group has no specific authority to the source physical file. More information on this parameter is in Commonly used parameters.

***LIBCRTAUT:** The public authority for the source physical file is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the source physical file). The public authority is determined when the source physical file is created. If the CRTAUT value for the library changes after the source physical file is created, the new value does not affect any existing objects.

***CHANGE:** The user can perform all operations on the source physical file except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the source physical file. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the source physical file.

***USE:** The user can perform basic operations on the source physical file, such as running a program or reading a file. The user cannot change the source physical file. *USE authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The user cannot access the source physical file.

authorization-list-name: Specify the name of the authorization list used.

TEXT Specifies the text that briefly describes the source physical file. More information on this parameter is in Commonly used parameters.

***BLANK:** Text is not specified.

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

Examples for CRTSRCPF

Example 1: Creating a File Without Members

```
CRTSRCPF   FILE(SRCLIB/PAYTXS)
```

This command creates a source file named PAYTXS in the SRCLIB library. The file is created without any members; therefore, no data can be put into the file until a member is added later. As many as 32,767 members (*NOMAX) can be added to the file.

Each member can have up to 10000 records before automatic extensions (499 increments maximum) occur that add 1000 records to the capacity of the member. Only minimum initial storage is allocated for each member with no restrictions on whether the space is connected. The public has object operational, read, add, delete, and update authority for the file, but no object management or object existence authority.

Example 2: Creating a File With a Member

```
CRTSRCPF   FILE(ORDERCTL/ORDERS)  MBR(*FILE)
  SIZE(100 50 5)
```

This command creates a source physical file named ORDERS in the ORDERCTL library. Storage space for the records placed in the file need not be contiguous. The initial allocation of storage provides for up to 100 records, and up to five increments of additional space for 50 records each can be added automatically. These allocation values also apply to members of this source file that will be added later.

Example 3: Creating a File that Contains DBCS Data

```
CRTSRCPF   FILE(IGCLIB/IGCSRC)  IGCDTA(*YES)
```

This command creates a source physical file named IGCSRC, which is stored in the library IGCLIB, and can contain DBCS data.

Error messages for CRTSRCPF

*ESCAPE Messages

CPF323C
QRECOVERY library could not be allocated.

CPF5702
File either not DDM file or not found.

CPF7302
File &1 not created in library &2.

CRTSPADCT (Create Spelling Aid Dictionary) Command Description

CRTSPADCT Command syntax diagram

Purpose

The Create Spelling Aid Dictionary (CRTSPADCT) command allows the user to create a user-defined spelling aid dictionary. This dictionary is used with the Create Document (CRTDOC), Check Document (CHKDOC), Edit Document (EDTDOC), or Work with Documents (WRKDOC) commands when doing spell check functions, such as spelling aid and spelling verification.

Spelling aid dictionaries are classified into IBM language dictionaries and user dictionaries. Language dictionaries are IBM-created; user dictionaries are created with this command.

The following names are the IBM-supplied system dictionary names, in alphabetical order, that are in library QDCT. It is recommended that they are not used for user-created dictionaries:

AFRIKAAN
AKTUEEL
BRASIL
CATALA
DANSK
DEUTSCH
DSCHWEIZ
ESPANA
FRANCAIS
FRA2
GREEK
ISLENSK
ITALIANO
LEGAL
MEDICAL
NEDERLND
NORBOK
NORNYN
PORTUGAL
SUOMI
SVENSK
UK
US

This command requires a source file that contains all the words to be put into the spelling aid dictionary. The source member is created by using the Start Source Entry Utility (STRSEU) command. The dictionary source type is SPADCT.

More information on user-defined spelling aid dictionaries is in the Using OfficeVision/400 Word Processing book.

Required Parameters

SPADCT

Specifies the qualified name of the spelling aid dictionary being created.

***USRSWL:** A user-defined stop word list is created using an IBM-supplied name.

The possible library values are:

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

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

spelling-aid-dictionary-name: Specify the name of the spelling aid dictionary being created.

SRCFILE

Specifies the qualified name of the source file used when the spelling aid dictionary is created. The source file contains the source member used for creating the dictionary.

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

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

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

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

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

Optional Parameters

SRCMBR

Specifies the name of the source file member that contains the words for the dictionary being created; the member is in the source file specified in the SRCFILE parameter. If SRCMBR is not specified, the member name is the same as that of the dictionary being created; the default value *DCT implies that the name of the dictionary being created is used. A member name must be specified when the source file member to be processed does not have the same name as the spelling aid dictionary being created.

***DCT:** The source file member name is the same as that of the dictionary being created.

source-file-member-name: Specify the name of the member in the source file (specified by SRCFILE) used to create the spelling aid dictionary.

OPTION

Specifies the type of output created when the dictionary is created.

Source Listing Options

***SRC or *SOURCE:** A list of the source statements used to create the dictionary, as well as any errors that occur, is created.

***NOSRC or *NOSOURCE:** No list of the source statements is created unless errors occur.

BASEDCT

Specifies the qualified name of the dictionary that contains words added to the dictionary being created. A system dictionary (one created by IBM) cannot be used here.

***NONE:** No words from another dictionary are added into the dictionary being created.

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

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

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

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

dictionary-name: Specify the name of the dictionary that contains words added to the dictionary being created.

VFYDCT

Specifies the qualified name of the dictionary that is searched for each word specified in the source member. Only words not found in the verify dictionary are placed in the new dictionary.

***NONE:** Use every word specified in the source member to create the spelling aid dictionary.

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

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

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

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

dictionary-name: Specify the name of the dictionary that contains words not included in the dictionary being created.

LNGATR

Specifies the language attribute associated with the dictionary being created. Because each language has its own structure, knowing the language for which the new dictionary is intended makes creation and storage of the new dictionary easier and more effective for the system.

***VFYDCT:** The language attribute of the dictionary is the same as the language attribute of the dictionary specified on the VFYDCT parameter.

***NONE:** The dictionary created has no specific language attribute.

***ENGLISH:** The language attribute of the spelling aid dictionary is English.

***ESPANA:** The language attribute of the spelling aid dictionary is Spanish.

***FRANCAIS:** The language attribute of the spelling aid dictionary is French.

***FRA2:** The language attribute of the spelling aid dictionary is French, in which accents are required on uppercase characters.

***ITALIANO:** The language attribute of the spelling aid dictionary is Italian.

***DANSK:** The language attribute of the spelling aid dictionary is Danish.

***DEUTSCH:** The language attribute of the spelling aid dictionary is German.

***ISLENSK:** The language attribute of the spelling aid dictionary is Icelandic.

***NEDERLND:** The language attribute of the spelling aid dictionary is Dutch.

***NORSK:** The language attribute of the spelling aid dictionary is Norwegian.

***SVENSK:** The language attribute of the spelling aid dictionary is Swedish.

***PORTUGAL:** The language attribute of the spelling aid dictionary is Portuguese.

***DSCHWEIZ:** The language attribute of the spelling aid dictionary is Swiss German.

***SUOMI:** The language attribute of the spelling aid dictionary is Finnish.

***CATALA:** The language attribute of the spelling aid dictionary is Catalan.

***AFRIKAAN:** The language attribute of the spelling aid dictionary is Afrikaans.

***GREEK:** The language attribute of the spelling aid dictionary is Greek.

***BRASIL:** The language attribute of the spelling aid dictionary is Brazilian Portuguese.

***TURKISH:** The language attribute of the spelling aid dictionary is Turkish.

***RUSSIAN:** The language attribute of the spelling aid dictionary is Russian.

SWLLANGID

Specifies the language identifier (ID) for the stop word list.

BASESWL

Specifies whether the IBM-supplied stop word list words are included in the user-created stop word list.

***IBM:** The words from the IBM-supplied stop word list are added to the user-created stop word list.

: No words from the IBM-supplied stop word list are added to the user-created stop word list.

Note: Word entries in the IBM-supplied stop word list source file preceded by a dash, (-), are not added to the user-created stop word list even when BASESWL(*IBM) is specified.

AUT Specifies the authority given to users who do not have specific authority to the spelling aid dictionary, who are not on an authorization list, and whose user group has no specific authority to the spelling aid dictionary. More information on this parameter is in Commonly used parameters.

***LIBCRTAUT:** The public authority for the spelling aid dictionary is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the spelling aid dictionary). The public authority is determined when the spelling aid dictionary is created. If the CRTAUT value for the library changes after the spelling aid dictionary is created, the new value does not affect any existing objects.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***USE:** The user can perform basic operations on the spelling aid dictionary, such as running a program or reading a file. The user cannot change the spelling aid dictionary. *USE authority provides object operational authority, read authority, and execute authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the spelling aid dictionary.

***EXCLUDE:** The user cannot access the spelling aid dictionary.

authorization-list-name: Specify the name of the authorization list used.

REPLACE

Specifies whether a dictionary being created has the same name as an existing dictionary. More information on this parameter is in Commonly used parameters.

***YES:** The dictionary being created replaces an existing dictionary that has the same name.

***NO:** No replacement occurs.

TEXT Specifies the text that briefly describes the spelling aid dictionary. More information on this parameter is in Commonly used parameters.

***SRCMBRTXT:** The text is taken from the source file member being used to create the spelling aid dictionary. If the source file is a database file, the text is taken from the source member. You can add or change text for a database source member by using the Start SEU (STRSEU), Add Physical File Member (ADDPFM), or Change Physical File Member (CHGPFM) commands. If the source file is an inline or device file, the text is blank.

***BLANK:** Text is not specified.

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

Example for CRTSPADCT

```
CRTSPADCT   SPADCT(MYLIB/MYDCT)  SRCFILE(MYLIB/SRC)
  SRCMBR(WORDS)  OPTION(*NOSRC)  BASEDCT(QGPL/BASDCT)
  VFYDCT(QDCT/US)  LNGATR(*ENGLISH)  REPLACE(*YES)
```

This command creates a spelling aid dictionary named MYDCT in the library MYLIB. The words used in the spelling aid dictionary are from source member WORDS of the SRC source file in MYLIB. The dictionary includes words from a dictionary named BASDCT, but does not contain any words that are found in the dictionary named US. The dictionary being created has the *ENGLISH attribute. If an existing dictionary is named MYDCT, it is replaced.

Error messages for CRTSPADCT

*ESCAPE Messages

CPF2283

Authorization list &1 does not exist.

CPF4102

File &2 in library &3 with member &4 not found.

CPF4104

User not authorized to operation on file &2 in &3, member, device, or program device &4.

CPF411B
Shared open of member &4 not successful.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9832
Function not supported for DDM file &2.

CPF9845
Error occurred while opening file &1.

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

CPF9899
Error occurred during processing of command.

CRTSQLPKG (Create Structured Query Language Package) Command Description

CRTSQLPKG Command syntax diagram

Purpose

The Create Structured Query Language Package (CRTSQLPKG) command is used to create (or re-create) an SQL package on a relational database from an existing distributed SQL program. A distributed SQL program is a program created by specifying the RDB parameter on a CRTSQLxxx (where xxx = C, CI, CPPI, CBL, CBLI, FTN, PLI, or RPG or RPGI) command.

Required Parameter

PGM Specifies the qualified name of the program for which the SQL package is being created. The program must be a distributed SQL program.

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

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

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

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

program-name: Specify the name of the program for which the package is being created.

Optional Parameters

RDB Specifies the name of the relational database where the SQL package is being created.

***PGM:** The relational database name specified for the SQL program is used. The relational database name is specified on the RDB parameter of the distributed SQL program.

relational-database-name: Specify the name of the relational database where the SQL package is to be created. Use the Work with Relational Database Directory Entry (WRKRDBDIRE) command to show the relational database names that are valid on this parameter.

USER Specifies the user name sent to the remote system when starting the conversation.

***CURRENT:** The user name associated with the current job is used.

user-name: Specify the user name being used for the application server job.

PASSWORD

Specifies the password to be used on the remote system.

***NONE:** No password is sent. If this value is specified, USER(*CURRENT) must also be specified.

password: Specify the password of the user name specified on the USER parameter.

GENLVL

Specifies the maximum severity level allowed for errors detected during SQL package creation. If errors occur at a level that exceeds the specified level, the SQL package is not created.

10: The default severity-level is 10.

severity-level: Specify the maximum severity level. Valid values range from 0 through 40.

REPLACE

Specifies whether an existing package is being replaced with the new package. More information on this parameter is in Commonly used parameters.

***YES:** An existing SQL package of the same name is replaced by the new SQL package.

***NO:** An existing SQL package of the same name is not replaced; a new SQL package is not created if the package already exists in the specified library.

DFTRDBCOL

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

***PGM:** The collection name specified for the SQL program is used. The default relational database collection name is specified on the DFTRDBCOL parameter of the distributed SQL program.

***NONE:** Unqualified names for tables, views, indexes, and SQL packages use the search conventions specified on the OPTION parameter of the CRTSQLxxx command used to create the program.

collection-name: Specify the collection name that is used for unqualified tables, views, indexes, and SQL packages.

PRTFILE

Specifies the qualified name of the printer device file to which the create SQL package error listing is directed. If no errors are detected during the creation of the SQL package, no listing is produced.

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

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

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

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

QSYSPRT: If a file name is not specified, the create SQL package error listing is directed to the IBM-supplied printer file QSYSPRT.

printer-file-name: Specify the name of the printer device file to which the create SQL package error listing is directed.

OBJTYPE

Specifies the type of program for which an SQL package is created.

***PGM:** Create an SQL package from the program specified on the PGM parameter.

***SRVPGM:** Create an SQL package from the service program specified on the PGM parameter.

MODULE

Specifies a list of modules in a bound program.

***ALL:** An SQL package is created for each module in the program. An error message is sent if none of the modules in the program contain SQL statements or none of the modules is a distributed module.

Note: CRTSQLPKG can process programs that do not contain more than 1024 modules.

module-name: Specify the names of up to 256 modules in the program for which an SQL package is to be created. If more than 256 modules exist that need to have an SQL package created, multiple CRTSQLPKG commands must be used.

Duplicate module names in the same program are allowed. This command looks at each module in the program and if *ALL or the module name is specified on the MODULE parameter, processing continues to determine whether an SQL package should be created. If the module is created using SQL and the RDB parameter is specified on the precompile command, an SQL package is created for the module. The SQL package is associated with the module of the bound program.

TEXT Specifies text that briefly describes the SQL package and its function.

***PGMTXT:** The text from the program for which the SQL package is being created is used.

***BLANK:** No text is specified.

'description': Specify a maximum of 50 characters of text, enclosed in apostrophes.

Example for CRTSQLPKG

```
CRTSQLPKG   PAYROLL  RDB(SYSTEMA)
  TEXT('Payroll Program')
```

This command creates an SQL package from the distributed SQL program PAYROLL on relational database SYSTEMA.

Error messages for CRTSQLPKG

***ESCAPE Messages**

SQL9004
Create of SQL package failed.

SQL9006
DB2 Query Manager and SQL Development Kit for AS/400 not at same install level as OS/400.

CRTSBSD (Create Subsystem Description) Command Description

CRTSBSD Command syntax diagram

Purpose

The Create Subsystem Description (CRTSBSD) command creates a subsystem description that defines the operational attributes of a subsystem. After the subsystem description is created, it can be specialized by commands that add, change, and remove work entries and routing entries in the subsystem description.

Restriction: The user of this command must have object operational authority and add authority for the library where the subsystem description is created.

Required Parameters

SBSD Specifies the qualified name of the subsystem description being created. The subsystem description is stored in the specified library.

More information on the IBM-supplied subsystem descriptions that are shipped with the system is in the Work Management  book.

The possible library values are:

***CURLIB:** The subsystem description is stored in the current library. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the subsystem description is stored.

subsystem-description-name: Specify the name of the subsystem description being created.

POOLS
Specifies one or more storage pool definitions that are in the subsystem description. Each definition designates one storage pool:

- Pool definition identifier: The identifier, inside the subsystem description, of the storage pool definition. The same identifiers can be used for pool definitions in different subsystem descriptions. Valid values range from 1 through 10.
- Size: The size of the storage pool expressed in kilobytes (1K = 1024 bytes). This is the amount of main storage that can be used by the pool.
- Activity level: The maximum number of threads that can run at the same time in the pool.

Up to 10 storage pool definitions can be specified for the subsystem description being created. Although each subsystem description can have as many as 10 storage pool definitions, there is an operational limitation on how many active storage pools can be in the *system*. In the system, no more than 64 storage pools can be active at one time, including the base storage pool and a machine storage pool. A storage pool for which *NOSTG has been specified is not considered active, and it is not allocated to any subsystem.

If a subsystem is started and all of its storage pools cannot be allocated without exceeding the 64-pool *system* maximum, the pools that can be allocated (up to the limit) are allocated and the

remainder are not. Then, for each routing step started by that subsystem that normally is routed into one of the pools that was not allocated, the base pool is used instead. More information about storage pools is in the Backup, Recovery, and Availability topic in the Information Center and the Work Management  book.

Element 1: Pool Identifier

pool-identifier: Specify the pool identifier of the storage pool definition to be in this subsystem. Valid values range from 1 through 10. The attributes of the pool must also be specified by one of the following values. As many as 10 sets of values can be specified in the POOLS parameter to define as many as 10 storage pools in the subsystem.

Element 2: Storage Size and Pool Definitions

***BASE:** The specified pool definition is defined to be the base system pool, which can be shared with other subsystems. The minimum size and activity level of the base pool are specified in the system values QBASPOOL and QBASACTLVL.

***NOSTG:** No storage and no activity level are assigned to the pool at first (the pool is inactive.)

***INTERACT:** The shared pool for interactive work.

***SPOOL:** The shared pool used for spooled writers.

***SHRPOOLnn:** The specified pool definition is defined to be a general-purpose shared pool. There are sixty general-purpose shared pools, identified by special values *SHRPOOL1 to *SHRPOOL60.

The size and activity level of a shared pool are specified using the Change Shared Storage Pool (CHGSHRPOOL) command.

storage-size: Specify the storage size (in kilobytes) of the specified storage pool. A value of at least 256 (meaning 256k) must be specified.

Element 3: Activity Level

activity-level: Specify the maximum number of threads that can run at the same time in the pool.

Optional Parameters

MAXJOBS

Specifies the maximum number of jobs allowed in the subsystem. The maximum applies to all jobs that are started and are waiting or running, except for jobs on the job queue or jobs that have finished running.

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

maximum-subsystem-jobs: Specify the maximum number of jobs allowed in the subsystem.

SGNDSPF

Specifies the qualified name of the sign-on display file that is used when showing sign-on displays at work stations allocated to the subsystem. If the sign-on display file does not exist when the subsystem is created, the user must specify a library qualifier because the sign-on display file name is kept. The sign-on display file must contain a record format by the name of SIGNON. More information is in the Application Display Programming  book.

***QDSIGNON:** The sign-on display file value QDSIGNON in QSYS is used when showing sign-on displays at work stations that are allocated to the subsystem.

The name of the sign-on display file can be qualified by one of the following library values:

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

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

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

display-file-name: Specify the name of the sign-on display file that is used.

SYSLIBLE
Specifies a library that is specified ahead of other libraries on the system library list. This parameter allows the user to create a secondary language library causing messages and displays to appear in the user's spoken language.

Note: The second language library cannot be specified in the QSYSLIBL and QUSRLIBL portions of the library list. To use this parameter, the QSYSLIBL library must contain fewer than 15 libraries. If the prior restrictions are not followed, a message is sent to the job log, but the second language library is not added to the library list.

***NONE:** A secondary language library is not used.

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

AUT Specifies the authority given to users who do not have specific authority to the subsystem description, who are not on an authorization list, and whose user group has no specific authority to the subsystem description.

***LIBCRTAUT:** The public authority for the subsystem description is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the subsystem description). The public authority is determined when the subsystem description is created. If the CRTAUT value for the library changes after the subsystem description is created, the new value does not affect any existing objects.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority. If the object is an authorization list, the user cannot add, change, or remove user ids.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the subsystem description.

***USE:** The user can perform basic operations on the subsystem description, such as displaying its contents. The user cannot change the subsystem description. *USE authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The user cannot access the subsystem description.

authorization-list-name: Specify the name of the authorization list used.

TEXT Specifies the text that briefly describes the subsystem description. More information on this parameter is in Commonly used parameters.

***BLANK:** Text is not specified.

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

Examples for CRTSBSD

Example 1: Creating a Description With a Signon Display File

```
CRTSBSD  SBSD(BAKER)  POOLS((1 *BASE)(2 2000 4))
  SGNDSPF(*LIBL/NEWSGNON)
  TEXT ('Subsystem for running Baker Department jobs')
```

This command creates a subsystem description named BAKER and stores it in the current library. If there is no current library, then it is stored in the general purpose library (QGPL). Storage pool definition 1 specifies that pool 1 is to share the base system pool; the definition of storage pool 2 is to have 2000K of storage and an activity level of 4. There is no limit in this subsystem description on the number of jobs that can be active at the same time. The activity levels in the subsystem may, however, be controlled by MAXACT parameters specified in work station entries, job queue entries, and routing entries that are in the subsystem. The sign-on display file is NEWSGNON and is used when showing sign-on displays at work stations allocated to the BAKER subsystem. The user's library list is searched for the NEWSGNON display file.

Example 2: Creating a Description that Contains Three Storage Pool Definitions

```
CRTSBSD  SBSD(MEDLIB/MEDICAL)
  POOLS((1 1500 2) (2 *BASE) (3 *NOSTG))
  MAXJOBS(5)  TEXT('Medical files
  Inquiry and update')
```

This command creates a subsystem description named MEDICAL and stores it in the MEDLIB library. The subsystem description contains three storage pool definitions: storage pool 1 is defined to have 1500K of storage and an activity level of 2, pool 2 is to share the base system pool, and pool 3 is defined first to be inactive when the other pools are active—it has no storage and no activity level. Up to five jobs can be active at the same time in this subsystem. A text description briefly describes the subsystem.

Error messages for CRTSBSD

***ESCAPE Messages**

CPF1696
Subsystem description &1 not created.

CRTS36DSPF (Create System/36 Display File) Command Description

CRTS36DSPF Command syntax diagram

Purpose

The Create System/36 Display File (CRTS36DSPF) command creates a display file from System/36 Screen File Generator (SFGR) source, and adds, deletes, or updates formats in existing display files.

Note: The data description specifications (DDS) source is saved in the DDS source file QS36DDSSRC, in the same library as the display file. If the QS36DDSSRC source file does not exist, this source file is created with a record length of 92.

Restriction: This command can be run either in the iSeries 400 or, when library QSSP is installed, in the System/36 environment. Using the TOFILE and TOMBR parameters, the user can convert the System/36 SFGR source to the iSeries 400 data description specifications (DDS) source.

Required Parameter

DSPFILE

Specifies the qualified name for the display file being created.

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

***CURLIB:** The display file is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the display file is created.

display-file-name: Specify the name of the display file being created.

Optional Parameters

OPTION

Specifies what to do with the specified display file. A display file can be created, or formats can be added, deleted, or updated in existing display files. Up to 32 options can be specified. A given option applies to the corresponding name in the SRCMBR parameter.

***CREATE:** A display file is created using the specified display file name as defined by all of the specified source members. When *CREATE is specified, no other options may be specified.

***ADD:** One or more display formats are added to the specified display file as defined by the corresponding names specified on the SRCMBR parameter. TGTRLS(*PRV) is invalid if OPTION(*ADD) is specified.

***UPDATE:** One or more display formats are updated in the specified display file as defined by the corresponding names in the SRCMBR parameter. TGTRLS(*PRV) is invalid if OPTION(*UPDATE) is specified.

***DELETE:** A display format is deleted from the specified display file. The format that is deleted is the name specified by the corresponding name specified in the SRCMBR parameter. If the deleted format was the only format in the display file, the entire display file is deleted.

SRCMBR

Specifies the member used in the source file when performing a create, add, or update operation. For the delete option, specify the name of the format being deleted.

***DSPFILE:** A member with the same name as the display file is used for the create, add, delete, or update option.

source-member-name: Specify the source member name that is used. Up to 32 source member names can be specified. The specified name is used for the corresponding option in the OPTION parameter. When the corresponding option is *DELETE, a format name *must* be specified instead of a member name.

SRCFILE

Specifies the qualified name of the source file in which the source members are located.

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

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

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

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

QS36SRC: The source file, QS36SRC, is used.

file-name: Specify the name of the source file that is used.

REPLACE

Specifies whether an existing display file is replaced. When OPTION(*CREATE) is not specified, this parameter is ignored. More information on this parameter is in Commonly used parameters.

***NO:** No replacement occurs.

***YES:** If a display file already exists, it is replaced. Other types of files are not replaced.

Note: A display file is not created if the display has the same name and library as an existing program or message file.

PRINT Specifies whether the compiler listings are printed.

***YES:** The full compiler listing for both the SFGR source syntax checking and CRTDSPF steps are printed.

***NO:** No compiler listing is printed for the SFGR source syntax checking step. *PARTIAL is assumed for the CRTDSPF step.

***PARTIAL:** If errors are found in the SFGR source syntax checking or CRTDSPF steps, the compiler listing is printed for that step, along with the error and warning messages. If no messages are issued, no compiler listing is printed.

MAXDEV

Specifies the number of devices that can use the display file at one time.

***SRCATR:** The maximum number of devices is taken from the first source member attributes (see the CHGS36SRCA or EDTS36SRCA commands). If the MAXDEV source attribute has not been set or is zero, a value of 5 is used.

number-of-devices: Specify the maximum number of devices that are permitted to use the display file. Valid values range from 1 through 256.

AUT Specifies the authority given to users who do not have specific authority to the System/36 display file, who are not on an authorization list, and whose user group has no specific authority to the System/36 display file.

***LIBCRTAUT:** The public authority for the System/36 display file is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the System/36 display file). The public authority is determined when the System/36 display file is created. If the CRTAUT value for the library changes after the System/36 display file is created, the new value does not affect any existing objects.

***USE:** The user can perform basic operations on the System/36 display file, such as running a program or reading a file. The user cannot change the System/36 display file. *USE authority provides object operational authority, read authority, and execute authority.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the System/36 display file.

***EXCLUDE:** The user cannot access the System/36 display file.

authorization-list-name: Specify the name of the authorization list used.

GENOPT

Specifies the options used to create the object.

***GEN:** The SFGR source syntax is checked unless SYNTAX(*NO) is specified. The SFGR source is converted to data description specifications (DDS), and an iSeries 400 display file is created. When GENOPT(*NOGEN) is specified, SYNTAX(*YES) must be specified.

***NOGEN:** A program object is not created. The compiler checks only the syntax of the statements.

***CONVERT:** The SFGR source syntax is checked unless SYNTAX(*NO) is specified, and the SFGR source is converted to data description specifications (DDS). A value should be specified for the TOMBR parameter to save the DDS.

When the to-member name is the same as that of the display file and the to-file is QS36DDSSRC, in the same library as the display file being created, the DDS is not saved in the specified source file member. Specify some other source file, library, or member name.

SYNTAX

Specifies whether the SFGR source syntax is checked.

***YES:** The source syntax is checked.

***NO:** The source syntax is not checked.

Note: If the SFGR source contains errors, the results are unpredictable.

TOFILE

Specifies the qualified name of the source file in which to store the DDS source that is used to create the display file. The to-file need not already exist. If it does not exist, the requester must be authorized to the Create Source Physical File (CRTSRCPF) command so it can be created. TOFILE is ignored if TOMBR(*NONE) is specified.

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

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

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

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

QDDSSRC: The source file, QDDSSRC, is used.

file-name: Specify the name of the source file in which to store the DDS source. The file should have a record length of 92 bytes.

TOMBR

Specifies the name of the source file member in which to store the data description specifications (DDS) source. If the member does not exist, it is created. When the member name is the same as that of the display file name, and the to-file is QS36DDSSRC in the same library as the display file being created, the DDS is saved in this member only if the compile operation of the display file is successful. To guarantee that the DDS is saved, specify the name of some other source file, library, or member.

***NONE:** The DDS source is not stored in the to-file. The source is saved in the Q36DDSSRC source file.

***SRCMBR:** The first name specified by the SRCMBR parameter is used as the member name.

***DSPFILE:** The display file name is used as the member name.

member-name: Specify the name of the source file member in which to store the DDS source. The member name is added if it does not already exist, and is replaced if it already exists. If any terminating errors are encountered, the member is not added or changed.

HALT Specifies whether an error message is issued when an error is found.

***YES:** The processing stops, and an error message is issued.

***NO:** The processing stops, and a diagnostic message is issued. The request ends, and a return code of 1008 is set.

DFRWRT

Specifies that the writing of data to the display file is delayed until a read request is made. Control is returned to the requesting program immediately after the data is received for output. This may result in improved performance.

***SRCATR:** The defer write option is taken from the DFRWTR attribute of the first source member (see the CHGS36SRCA or EDTS36SRCA commands). If there is no source member, or if the DFRWTR source attribute has not been set, *YES is assumed.

***YES:** When a write request is made to the display file, control is returned after the buffer is processed. The data may not be displayed immediately; the display of the data may take place later when a read or combined write/read operation is performed. The buffer is then available for the next read or combined write/read operation.

***NO:** After a write operation to the display file, the entire input/output request is completed before returning control to the requesting program. All input/output feedback information is available.

TGTRLS

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

When specifying the *target-release* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V3R6M0 is version 3, release 6, modification level 0.

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

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

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

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

Example for CRTS36DSPF

```
CRTS36DSPF  DSPFILE(MYLIB/MYDISPLAY)  OPTION(*CREATE)
  SRCMBR(SFGRMBR1 SFGRMBR2)  SRCFILE(MYLIB/QS36SRC)
  GENOPT(*NOGEN)
```

This command checks the SFGR source (producing a printer list named QPUTSFGR). No DDS is built, and the display file is not created.

Error messages for CRTS36DSPF

*ESCAPE Messages

SSP4464
Member &3 in file &1 in use, cannot be shared.

SSP5003
Source member &1 not found.

SSP5004
&1—This load member exists, but is not a $SFGR member.

SSP5005
&1 display file already exists.

SSP5007
Source member &1 already given.

SSP5009
Display file &1 not found in library &2.

SSP5010
&1 not System/36 display file.

SSP5011
&1 not allowed for display file name.

SSP5012
Format &1 not found in display file

SSP5015
Source file library &1 not found.

SSP5016
Display file library &1 not found.

SSP5017
TOFILE library &1 not found.

SSP5019
Terminating errors in $SFGR input specifications.

SSP5027
TGTRLS(*PRV) allowed with changes only when existing display file created for previous release.

SSP5451
Existing file &1 is not a display file.

SSP6124
Unexpected error occurred.

SSP7375
Error &1 received by &2 utility.

SSP8663
User not authorized to access &1.

SSP8679
Not authorized to access member &1.

SSP9080
Object &1 in use; it cannot be shared.

CRTS36MNU (Create System/36 Menu) Command Description

CRTS36MNU Command syntax diagram

Purpose

The Create System/36 Menu (CRTS36MNU) command creates a menu (display file and command message file) from the user's source members. The user can specify that this menu be created either in fixed-format, with options 1 through 24 arranged in two columns, or in free-format.

Restriction: The CRTS36MNU command can be run either in the System/36 environment, or when the QSSP product library is installed, on the iSeries 400.

Required Parameter

CMDTXTMBR
Specifies the source member that contains the System/36 message source used to create the command text message file. The command text message file contains the text for the command that is run when the option is selected for that command. The message IDs must begin with USR. Trailing ## symbols are required by a System/36 on the CMDTXTMBR name. The menu (display file) name is determined by removing the ## symbols.

Note: The message file name specified within the source member must be the same as the source member name.

Optional Parameters

OPTTXTMBR
Specifies the source member used to describe the options on the menu the user is creating.

***NONE:** The option text is taken from the command text message file. If OPTTXTMBR(*NONE) is specified, FREEFORM(*NO) must also be specified.

member-name: Specify the member name of the source to use for the descriptions of the options on the menu the user is creating.

The member may contain either the System/36 SFGR source or the System/36 message source. If the member name is menu-name DT (the name obtained by removing the ## symbols from the end of the CMDTXTMBR name), the member must contain the System/36 message source. If the member name is the same as the menu name, then the member must contain the System/36 SFGR source. When any other name is specified, the contents of the member is examined to determine the type of source being used. A source member must be specified when FREEFORM(*YES) is specified.

Note: The message file name specified in the source member must be the same as the source member name.

CMDTXTSRC

Specifies the qualified name of the source file in which the CMDTXTMBR is located.

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

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

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

QS36SRC: The source file, QS36SRC, is used.

file-name: Specify the name of the source file that contains the member specified by the CMDTXTMBR parameter.

OPTTXTSRC

Specifies the qualified name of the source file where the OPTTXTMBR is located.

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

***CMDLIB:** The library specified in CMDTXTSRC is used to locate the file.

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

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

QS36SRC: The source file, QS36SRC, is used.

file-name: Specify the name of the source file that contains the member specified by the OPTTXTMBR parameter.

MNULIB

Specifies the menu library where both the menu display file and command text message file are created.

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

***CMDLIB:** The menu is created in the command text source (CMDTXTSRC) library.

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

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

REPLACE

Specifies whether an existing display file is replaced. If terminating errors are encountered, the

existing display file is not replaced. If the display file already exists, the AUT parameter is ignored, and the authorities for the old display file are copied to the new display file that replaces it. More information on this parameter is in Commonly used parameters.

***NO:** No replacement occurs.

***YES:** An existing display file is replaced by the new one being created. Other types of files are not replaced.

Note: A menu is not created if it has the same name and library as an existing program or message file.

FREEFORM

Specifies whether the menu is created in free format or in a fixed format, with two columns each containing twelve items.

***NO:** Free-format is not used. The menu is created using a fixed-format with two-columns. The option text message numbers correspond to the option number. The system supplies the option numbers displayed on the menu.

***YES:** The menu is created using free-format. OPTTXTMBR is required when FREEFORM(*YES) is specified. Message numbers correspond to the row numbers on the screen. FREEFORM(*YES) is required when the OPTTXTMBR contains the screen file generator (SFGR) source. The user's OPTTXTMBR source must supply the option numbers shown on the menu.

KEEP Specifies whether the option text message file is kept when the compilation is complete.

***NO:** The option text message file is not kept.

***YES:** The option text message file is kept. If the option text source member contains the SFGR source and has the same name as the menu, then KEEP(*YES) is required.

DDSLIST

Specifies whether a full or partial DDS compiler listing is provided.

***PARTIAL:** A partial compiler listing is provided.

***FULL:** A full DDS compiler listing and cross-reference are provided.

MAXDEV

Specifies the maximum number of devices that can use the menu at one time.

5: The maximum number of devices is five.

number-of-devices: Specify the maximum number of devices that can use the menu at one time. Possible values range from 1 through 256.

AUT Specifies the authority given to users who do not have specific authority to the System/36 menu, who are not on an authorization list, and whose user group has no specific authority to the System/36 menu.

***LIBCRTAUT:** The public authority for the System/36 menu is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the System/36 menu). The public authority is determined when the System/36 menu is created. If the CRTAUT value for the library changes after the System/36 menu is created, the new value does not affect any existing objects.

***USE:** The user can perform basic operations on the System/36 menu, such as running a program or reading a file. The user cannot change the System/36 menu. *USE authority provides object operational authority, read authority, and execute authority.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the System/36 menu.

***EXCLUDE:** The user cannot access the System/36 menu.

authorization-list-name: Specify the name of the authorization list used.

TOFILE

Specifies the qualified name of the source file (to-file) in which to store the data description specifications (DDS) source that was used to create the object. If the file does not exist, the requester must be authorized to the Create Source Physical File (CRTSRCPF) command so it can be created. This parameter is ignored if TOMBR(*NONE) is specified.

Note: See special case for TOMBR when QS36DDSSRC is specified.

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

***MNULIB:** The menu library is used to locate the to-file.

***CMDLIB:** The command library is used to locate the to-file.

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

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

QDDSSRC: The source file, QDDSSRC, is used.

file-name: Specify the name of the source file in which to store the DDS source. The file should have a record length of 92 bytes.

TOMBR

Specifies the name of the source file member in which to store the data description specifications (DDS) source. If the member does not exist, it is created. When the member name is the same as that of the display file name, and the to-file is QS36DDSSRC in the same library as the display file being created, the DDS is saved in this member only if the compile operation of the display file is successful. To guarantee that the DDS is saved, specify the name of some other source file, library, or member.

***NONE:** The DDS source is not stored in the to-file.

member-name: Specify the name of the source file member in which to store the DDS source. The member is added if it does not already exist, and replaced if it already exists.

IGCDTA

Specifies whether the file contains double-byte character set (DBCS) data.

***NO:** The file does not process DBCS data.

***YES:** The file processes DBCS data.

TGTRLS

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

When specifying the *target-release* value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V3R6M0 is version 3, release 6, modification level 0.

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

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

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

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

Example for CRTS36MNU

```
CRTS36MNU  CMDTXTMBR(MENU##)  OPTTXTMBR(SFGRMBR)
  CMDTXTSRC(MYLIB/QS36SRC)  OPTTXTSRC(MYLIB/QS36SRC)
  MNULIB(WORKLIB)  REPLACE(*YES)  FREEFORM(*YES)
  KEEP(*YES)
```

This command creates a free format menu display from the specified SFGR source member, placing the resulting menu into library WORKLIB. Since the SFGR source is being used, FREEFORM(*YES) and KEEP(*YES) are required. The layout of the menu display is controlled by the format definition located in the SFGR source member. No special checking is done to ensure that the display file defined by this source is valid when used by the system as a menu.

Error messages for CRTS36MNU

*ESCAPE Messages

SSP4464

Member &3 in file &1 in use, cannot be shared.

SSP5004

&1—This load member exists, but is not a $SFGR member.

SSP5005

&1 display file already exists.

SSP5011

&1 not allowed for display file name.

SSP5017

TOFILE library &1 not found.

SSP5019

Terminating errors in $SFGR input specifications.

SSP5027

TGTRLS(*PRV) allowed with changes only when existing display file created for previous release.

SSP5451
Existing file &1 is not a display file.

SSP5750
Command message file messages 1-24 contain only blank text.

SSP5751
Command text message file name must end with ##.

SSP5755
Unable to create $BMENU work file.

SSP5756
Command message file name must be longer than 2 characters.

SSP5758
Command text source member &1 not found.

SSP5759
Option text source member &1 not found.

SSP5760
Command text message file has wrong name.

SSP5761
Option text message file has wrong name.

SSP5763
Option text member name should not be same as menu name.

SSP5766
Option text source library &1 was not found.

SSP5767
Menu library &1 not found.

SSP5768
Command text source library &1 not found.

SSP5772
Option text source member required for free format menu.

SSP5773
Command and option text member names must not be the same.

SSP6124
Unexpected error occurred.

SSP7375
Error &1 received by &2 utility.

SSP8663
User not authorized to access &1.

SSP8679
Not authorized to access member &1.

SSP9080
Object &1 in use; it cannot be shared.

CRTS36MSGF (Create System/36 Message File) Command Description

CRTS36MSGF Command syntax diagram

Purpose

The Create System/36 Message File (CRTS36MSGF) command creates a message file from the user's System/36 message source member. Using the TOFILE and TOMBR parameters, the user can convert the System/36 message source to an AS/400 system message source.

Restrictions: When the QSSP product library is installed, the CRTS36MSGF command can be run either in the System/36 environment or on the AS/400 system.

System/36 message source consists of 3 types of statements:

1. The message control statement must be the first record in the source. Only one control statement is allowed. The control statement has the following syntax:

   name<,level> <comment>

   where: name is the name of the message file being created or changed. The name must start in column 1 of the source record. Level must be a 1 or 2 or a blank, and is separated from the name by a comma. A level 1 indicates that the source is for message text (first-level messages); a level 2 indicates that the source is for online help information (second-level messages). If level is omitted or blank, a first level message is assumed. Anything after the first blank in the control statement is considered to be a comment and is ignored.
2. Comment records that start with an asterisk (*) in column 1. All comment records are ignored.
3. One or more message text statements may be defined. The message text statement has the following syntax:

   mmmm text

   where: mmmm is the System/36 Message Identification Code (MIC) and must start in column 1 and consist of 4 digits (0-9). All 4 digits are required. The 7-character message identifier is created by adding the message prefix specified by the MSGPFX parameter in front of the MIC. Column 5 is ignored and should be left blank. Text is the text of the message being defined and starts in column 6. The message text statements must be arranged so that the MICs are not in descending order. If there is insufficient room to define the entire message in a single record, the MIC can be repeated in column 1 of the next record and the text can be continued starting in column 6.

   A first level message is restricted to a maximum of 75 characters, and a second level message is restricted to a maximum of 225 characters unless RESTRICT(*NO) is specified. Trailing blanks on the last record for each MIC are not counted. The record length used when processing the SRCMBR is the RCDLEN source attribute for that member. This attribute defaults to the source file record length, but can be set or changed when using the CHGS36SRCA, EDTS36SRCA, or RSTS36LIBM commands. The RCDLEN source attribute is the member's logical record length. If the RCDLEN source attribute is less than the record length of the source file, all characters after the logical record length are ignored. If any record contains any non-blank characters after the logical record length, a diagnostic message is issued as a warning.

When SUBST(*YES) is specified on this command, certain strings of # symbols within the message text are treated as substitution fields. A # symbol string (a series of 1 or more characters defined by code point '7B'X) qualifies as a substitution field only if it is preceded and followed by one of the following delimiter characters:

Char	Hex	Description
	40	Blank
	41	Blank
\	E0*	Backslash
<	4C	Less Than
(	4D	Left Parenthesis
+	4E	Plus
|	4F*	Vertical Bar
!	4F*	Exclamation
&	50	Ampersand
*	5C	Asterisk

Char	Hex	Description
)	5D	Right Parenthesis
;	5E	Semicolon
¬	5F*	Not
^	5F*	Caret
-	60	Minus
/	61	Slash
,	6B	Comma
%	6C	Percent
>	6E	Greater Than
?	6F	Question
:	7A	Colon
’	7D	Quote
=	7E	Equals
“	7F	Double Quote

Note: Some hex code points X’4F’ and X’5F’ are assigned different graphics. The comparison is for the HEX value, not the graphic entered. The backslash (’E0’X) is not a CL delimiter, but is also not allowed in a name. No character below ’40’X is allowed in a name.

The delimiter that precedes the # symbol string need not be the same as the delimiter that follows the # symbol string. If a # symbol string occurs at the beginning or end of the message text, it is treated as if it were preceded or followed by a blank, respectively.

Double-Byte Character Set Considerations:

If any double-byte character set (DBCS) text is contained in a message, the first MIC for the message must be preceded by a record containing an MIC of A000. This special MIC resets the current MIC counter so that the non-descending MIC requirement is temporarily suspended, allowing the user to define DBCS versions of previously defined MICs within the same message file. The special A000 MIC also alters the third character of the MSGPFX, changing it to a ’Z’. If DBCS messages will be defined, the user should not specify a MSGPFX that uses ’Z’ for the third character or duplicate message ID errors may result.

Required Parameter

SRCMBR

Specifies the source member that is used to create the message file. The first record that is not a comment record in the source member specifies the name of the message file being created or changed. The first record that is not a comment record also specifies whether the text in this source member is for the first-level or second-level message text.

Optional Parameters

SRCFILE

Specifies the qualified name of the source file used to create the message file.

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

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

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

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

QS36SRC: The source file named QS36SRC is used.

file-name: Specify the name of the source file that contains the member specified by the SRCMBR parameter.

MSGLIB

Specifies the qualified name of the library used to receive the message file that is created.

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

***CURLIB:** The message file is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

***SRCLIB:** The source library is used to locate the message file.

library-name: Specify the name of the library where the message file is created.

REPLACE

Specifies whether an existing message file is replaced. This parameter is ignored if OPTION(*CREATE) is not specified. More information on this parameter is in Commonly used parameters.

***NO:** No replacement occurs.

***YES:** If a message file already exists, it is replaced by the one being created.

Note: A message file is not created if the message file has the same name and library as an existing program, display file, or other type of file.

MSGPFX

Specifies the message prefix used to create messages. The default prefix of USR must be used if the message is used to build a menu.

USR: The message prefix of USR is used.

prefix-name: Specify the three-character message prefix used in message IDs.

OPTION

Specifies what to do with the specified message file. A message file is created, or messages are added or changed, in existing message files.

***CREATE:** A message file is created by using the message file name specified in the source.

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

***CHANGE:** Messages are updated in the specified message file. If a given message does not exist, it is added.

SUBST

Specifies that System/36 text replacement fields are converted into the notation that represents text replacement fields in AS/400 system message files. Although *YES is the default, not everyone uses text replacement fields and *NO is a way to make sure that a field is not converted by mistake. Replacement fields are represented by a pair of trailing ## symbols in the S/36 source.

***YES:** The text replacement fields are converted.

***NO:** The text replacement fields are not converted.

RESTRICT

Specifies that all System/36 restrictions on message text length are enforced.

***YES:** System/36 restrictions are enforced on the message text length. First-level message text is limited to 75 characters. Second-level message text is limited to 225 characters.

***NO:** System/36 restrictions on the message text length are not enforced.

AUT Specifies the authority given to users who do not have specific authority to the System/36 message file, who are not on an authorization list, and whose user group has no specific authority to the System/36 message file. More information on this parameter is in Commonly used parameters.

***LIBCRTAUT:** The public authority for the System/36 message file is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the System/36 message file). The public authority is determined when the System/36 message file is created. If the CRTAUT value for the library changes after the System/36 message file is created, the new value does not affect any existing objects.

***USE:** The user can perform basic operations on the System/36 message file, such as running a program or reading a file. The user cannot change the System/36 message file. *USE authority provides object operational authority, read authority, and execute authority.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the System/36 message file.

***EXCLUDE:** The user cannot access the System/36 message file.

authorization-list-name: Specify the name of the authorization list used.

TOFILE

Specifies the qualified name of the source file (to-file) in which to store the data description specifications (DDS) source that was used to create the object. If the file does not exist, the requester must be authorized to the Create Source Physical File (CRTSRCPF) command so it can be created. This parameter is ignored if TOMBR(*NONE) is specified.

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

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

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

QCLSRC: The source file QCLSRC is used.

file-name: Specify the source file name under which to store the source file.

TOMBR

Specifies the name of the source file member in which to store the data description specifications (DDS) source. If the member does not exist, it is created. When the member name is the same as

that of the display file name, and the to-file is QS36DDSSRC in the same library as the display file being created, the DDS is saved in this member only if the compile operation of the display file is successful. To guarantee that the DDS is saved, specify the name of some other source file, library, or member.

***NONE:** The CL source is not stored in the to-file.

***SRCMBR:** If no member name is specified, the member name specified in the SRCMBR parameter is used.

member-name: Specify the name of the source file member in which to receive the CL source file.

HALT Specifies whether the processing stops when an error is found in the source.

***YES:** The processing stops and an error message is sent.

***NO:** The request ends and a return code of 2034 is set. A diagnostic message is sent.

***IGNORE:** A diagnostic message is sent and the current source statement is ignored; processing continues with the next source statement.

Example for CRTS36MSGF

```
CRTS36MSGF  SRCMBR(MYMSGSRC)  SRCFILE(MYLIB/QS36SRC)
  MSGLIB(WORKLIB)  REPLACE(*YES)
```

This command creates a message file named MYMSGSRC in library WORKLIB. If a message file with the same name already exists, this new message file replaces the existing one.

Error messages for CRTS36MSGF

***ESCAPE Messages**

SSP1727
Message file &1 not found in library &2.

SSP5017
TOFILE library &1 not found.

SSP6124
Unexpected error occurred.

SSP7375
Error &1 received by &2 utility.

SSP8663
User not authorized to access &1.

CRTTBL (Create Table) Command Description

CRTTBL Command syntax diagram

Purpose

The Create Table (CRTTBL) command creates a named table. The table can be used for the translation of data being moved between the system and a device (a display, for example). The table can also be used to specify an alternative collating sequence or to specify field translation functions.

You can create a table by prompting for the source information, or by specifying a source member.

To create a table using prompting support, specify *PROMPT on the SRCFILE parameter. A display is shown allowing you to view and change the table values and a new table is created based on the values chosen on the display.

To create a *CVT table using a source member, you must specify a source member that contains 8 records. Each record in this member must contain 64 hexadecimal characters (characters after position 64 are not used). This input (512 hexadecimal characters) is converted and stored internally as 256 bytes by the CRTTBL command.

The value you specify for a position within the source member is the same value that is returned by QDCXLATE (or other system program) whenever that position is encountered.

For example, if you specify “C0C1C2C3C4C5C6...’’ as the first part of record 1 in the source, then a hexadecimal ”C0“ is returned when given a hexadecimal ”00“, a hexadecimal ”C1“ is returned when given a hexadecimal ”01“ and so on.

To create a *UCSSRTSEQ table using a source member, you must specify a source member that contains the following layout of information.

1. column 1-4 = Hex code point to be sorted
2. column 6-10 = Weight of this code point as a decimal number
3. column 11-80 = not used, can contain such things as comments.

Required Parameter

TBL Specifies the qualified name of the table being created.

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

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

library-name: Specify the name of the library where the table is created.

table-name: Specify the name of the table being created.

Optional Parameters

SRCFILE

Specifies the qualified name of the source file that contains the description of the table being created or that prompting support is to be used.

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

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

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

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

QTBLSRC: The system source file named QTBLSRC contains the source records that are used with this command to create the table.

source-file-name: Specify the name of the source file that contains the source records that are used with this command to create the table.

Other Single Value

***PROMPT:** Allows the user to access and use source information.

SRCMBR
Specifies the name of the source file member that contains the description of the table being created.

***TBL:** The source file member name is the same as the name of the table.

source-file-member-name: Specify the name of the member in the source file specified on the SRCFILE parameter that is used to create the table.

TBLTYPE
Specifies the type of table object to be created.

***CVT:** A conversion type table is created.

***SRTSEQ:** A sort sequence type table is created.

***UCSSRTSEQ:** An ISO-10646 UCS-2 sort sequence type table is created.

BASETBL
Specifies the base table to be used for prompting support when creating a conversion table.

***HEX:** A one to one mapping table is used.

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

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

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

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

table-name: Specify a table object to be used as a base.

BASESRTSEQ
Specifies the base table to be used for prompting support when creating a sort sequence table.

***JOB:** The sort sequence must be resolved when the object is created.

***LANGIDSHR:** A shared weight sort table is used.

***LANGIDUNQ:** A unique weight sort table is used.

***HEX:** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence. The CCSID for hexadecimal data is 65535.

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

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

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

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

table-name: Specify the name of a table object to be used as a base.

BASELANGID

Specifies the base language used for prompting support when creating a sort sequence table.

***JOB:** The sort sequence must be resolved when the object is created.

language-ID: Specify a valid language ID to be used for the sort sequence table being created.

CCSID

Specifies the coded character set ID (CCSID) in which to store the sort sequence table information.

***JOB:** The sort sequence must be resolved when the object is created.

coded-character-set-ID: Specify the CCSID to be used for the sort sequence table information.

AUT Specifies the authority given to users who do not have specific authority to the table, who are not on an authorization list, and whose user group has no specific authority to the table. More information on this parameter is in Commonly used parameters.

***LIBCRTAUT:** The public authority for the table is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the table). The public authority is determined when the table is created. If the CRTAUT value for the library changes after the table is created, the new value does not affect any existing objects.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the table.

***USE:** The user can perform basic operations on the table, such as running a program or reading a file. The user cannot change the table. *USE authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The user cannot access the table.

authorization-list-name: Specify the name of the authorization list used.

TEXT Specifies the text that briefly describes the table. More information on this parameter is in Commonly used parameters.

***BLANK:** Text is not specified.

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

Example for CRTTBL

```
CRTTBL   TBL(SCRAMTBL)  SRCFILE(USERTABLES)
  SRCMBR(SCRAMBLE)  TEXT('Translate table for
  scrambling text characters')
```

This command creates a table named SCRAMTBL and (by default) stores it in the current library. The source file named USERTABLES contains the source records used when the table is created; the name of the source file member is SCRAMBLE. The TEXT parameter describes this table as being used as a translate table for scrambling text characters.

Error messages for CRTTBL

***ESCAPE Messages**

CPF2614
Table &1 in &2 already exists.

CPF2623
Library &1 not found.

CPF2678
Data in source file &1 in &2 not valid.

CPF3BF7
Data in source file at line &1 is not correct

CPF3FC9
Value &1 for CCSID not valid.

CRTTAPCGY (Create Tape Category) Command Description

CRTTAPCGY Command syntax diagram

Purpose

The Create Tape Category (CRTTAPCGY) command creates a user defined category name and assigns it to a system name. A category can be used to group volume identifiers together. The following special value categories have been provided for use by the OS/400 system and are already defined for all library devices.

- *NOSHARE

  The volume identifier is only used by the system that owns the rights of the *NOSHARE. Note that random access cartridge loader (RACL) devices in RACL mode can have cartridge identifiers in the *NOSHARE category, but the security of the *NOSHARE is not handled by the OS/400 system.
- *SHARE400

  The cartridge identifier can be shared by all iSeries 400 that are attached to the library device.
- *IPL

  This category should be used for cartridge identifiers that would be used in an alternate IPL. The management of the cartridges in the category must be done by the user.
- *NL

  The cartridge identifer in this category must have no logical volume identifier in order for it to be used. If it has a logical volume identifier, an Initialize Tape (INZTAP) command must be done to initialize the tape volume to a non-labeled tape before it can be used.
- *INSERT

  The cartridge identifier has been placed in the library device, but has not yet been added to the system. An Add Tape Cartridge (ADDTAPCTG) command must be done before any I/O can be done to the tape volume.
- *EJECT

  The volume identifer has been removed from the system by a Remove Tape Cartridge (RMVTAPCTG) command and is no longer usable by the library device.
- *CNV

  When a tape in this category is unloaded by specifying ENDOPT(*UNLOAD), OS/400 will automatically export the tape to the convenience station.
- *SYSGEN

  The cartridge ID exists in the *SYSGEN category. *SYSGEN category is used for all cartridges when the library device description is in *SYSGEN mode. A library device description is in *SYSGEN mode when the GENCTGID parameter is *SYSGEN in the device description. *SYSGEN mode is used to

generate identifiers for non-bar code libraries instead of loading and reading all the logical volume identifiers directly from the tape. If the library device is in *SYSGEN mode, cartridges cannot be moved from the *SYSGEN category.

Required Parameter

CGY Specifies the category being created.

Element 1: Category Name

category-name: Specify the name of the category to create.

Element 2: Category System

This element identifies the system to which the category belongs. The system name is obtained from the current system name field of a Display Network Attributes (DSPNETA) command.

***CURRENT:** The category belongs to the system currently running the command.

system-name: Specify the name of the system to which the category belongs.

Note: In order to specify an owning system other than *CURRENT, the category must have been defined previously on the system. For example, if system A and system B are attached to library device LIB01, then CRTTAPCGY CGY(CAT1 A) creating category CAT1 and assigning as its owner system A is required on system A before a CRTTAPCGY CGY(CAT1 A) creating category CAT1 and assigning as its owner system A can be done on system B. If both of these create commands are successful, CAT1 owned by system A is logically considered the same category and can be used for cartridges in library LIB01.

Example for CRTTAPCGY

```
CRTTAPCGY   CGY(CAT1 RCHAS215)
```

This command creates a user defined category named CAT1 and assigns as its primary owner system RCHAS215.

Error messages for CRTTAPCGY

***ESCAPE Messages**

CPF67DD
Category not created.

CPF67E2
Category already exists

CRTTAPF (Create Tape File) Command Description

CRTTAPF Command syntax diagram

Purpose

The Create Tape File (CRTTAPF) command creates a tape device file. The device file contains the file description, which identifies the device to be used; it does not contain data. The tape device file is used to read and write records on tape. The same device file can be used for both input and output operations.

Note: This command is not used to create device files for use in save and/or restore operations. User-created device files are not needed for save and/or restore operations.

Tape files have no data description specifications (DDS). The information in the tape file description comes from the command that creates it. The tape file has one record format for input/output operations. The record format consists of one character field containing the input data retrieved from the device or the output data to be written to the device. The program using the device file must describe the fields in the record format so the program can arrange the data received from or sent to the device in the manner specified by the tape file description.

Required Parameter

FILE Specifies the qualified name of the file being created. If the file is used by a high-level language (HLL) program, the file name must be consistent with the naming rules of that language; otherwise, the file must be renamed in the program.

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

***CURLIB:** The tape device file is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the tape device file is created.

tape-device-file-name: Specify the name of the tape device file being created.

Optional Parameters

DEV Specifies the names of one or more tape devices or one media library device used with this tape device file to perform input and output data operations. A media library device is a tape storage device that contains one or more tape drives, tape cartridges, and a part (carriage and picker assembly) for moving tape media between the cartridge storage slots and the tape drives.

***NONE:** No device name is specified. The name of the tape device must be specified later in a CHGDSPF or OVRDSPF command, or in the HLL program that opens the file.

device-name: Specify the names of one or more devices (no more than four) or the name of one media library device used with this tape device file. The order in which the device names are specified is the order in which tapes on the devices are processed. When more volumes are processed than the number of devices in the DEV list, the devices are used in the same order specified, wrapping around to the first device as needed. Each device name must be known on the system by a device description before this device file is created.

VOL Specifies one or more volume identifiers used by the file. The volumes must be installed in the same order as the identifiers are specified here (and as they are specified on the DEV parameter). If the file is opened for read backward, then the volume identifiers in the list are processed from last to first (while the devices in the device list are used in first-to-last order). If a list of volume identifiers is provided for the file, operator messages indicate the name of the required volume. More information on this parameter is in Commonly used parameters.

***NONE:** No tape volume identifiers are specified for this file. They can be supplied before the device file is opened, either in a CHGTAPF or OVRTAPF command or in the high-level language program. If volume identifiers are not specified before the device file is opened, volume checking is not performed beyond verifying that the correct label type volume is on the device, and volume

names are not provided in operator messages. The maximum number of reels processed for an *NL, *NS, *BLP, or *LTM input file when VOL(*NONE) is specified is determined by the REELS(number-of-reels) parameter value.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are placed on the device. Each volume identifier contains a maximum of 6 alphanumeric characters. Use a blank as a separator character when listing multiple identifiers. Up to 50 volume identifiers can be specified. These identifiers are used in messages sent to the operator during processing. The maximum number of reels processed for an *NL, *NS, *BLP, or *LTM input file is determined by the number of volume identifiers in the list.

Note: If the VOL parameter value used for the file specifies a list of identifiers rather than VOL(*NONE), the number-of-reels part of the REELS parameter is ignored regardless of where it is specified. A description of how the parameter values for the file are determined when overrides are used, the high-level language interface, and the device file when the file is opened is in the File Management topic in the Information Center. To ensure that the number-of-reels part of the REELS parameter is used (rather than a VOL identifier list) to control the volumes processed by the tape device file, specify VOL(*NONE) in the same command in which the REELS parameter is specified.

REELS

Specifies the type of labeling used on the tape reels and the maximum number of reels processed if both a list of volume identifiers is not specified (VOL parameter) and this device file is used with either *NL, *NS, *LTM, or *BLP input files. When the number of reels is specified as the second element of this parameter, the volume identifiers on the volumes are ignored if labeled tapes are being processed; instead, the order in which the reels are installed on the device must be checked by the operator.

The number-of-reels value is not a limiting value for standard-label or output files. For a standard-label *input* file, the data file labels limit the number of volumes processed by indicating end-of-file. For an *output* file, the number-of-reels value is ignored; the system requests that additional volumes be kept on the device until the file is closed.

The system checks the first record following the load point on the tape to see (1) whether it has exactly 80 bytes for EBCDIC or at least 80 bytes for ASCII and (2) whether the first 4 bytes contain the values VOL and 1. If so, the reel contains a standard-label tape. *SL and *BLP files require standard-label tape volumes. *NL, *NS, and *LTM tape files cannot process standard-label volumes.

Note: The values *SL, *NL, and *LTM can be specified if the device file is used for either reading or writing on tapes. The values *NS and *BLP are valid only if the device file is used to read tapes.

Element 1: Type of Labels

***SL:** The volumes have standard labels. If a list of volume identifiers is specified (with the VOL parameter), the system checks that the correct tape volumes are on the device in the specified sequence.

- If no volume identifier list is given and the file is opened for *output*, any standard-label volumes may be installed on the device.
- If no volume identifier list is given and the file is opened for *input*, the first volume may have any volume identifier, but if the file is continued, the system requires the correct continuation

volumes to be processed (verified by checking the data file labels). For an input file, the end-of-file message is sent to the program being used when the labels on the last volume processed indicate that it is the last volume for the data file.

***NL:** The volumes are not labeled. On a nonlabeled volume, tape marks are used to indicate the end of each data file and the end of the volume. For an *input* file, the end-of-file message is sent to the program when the number of volumes specified in the volume list have been processed, or, if no list of volume identifiers is provided, when the number of reels specified in the REELS parameter are processed.

***NS:** The volumes have nonstandard labels. Each volume must start with some kind of label information, optionally preceded by a tape marker and always followed by a tape marker. This nonstandard label information is ignored. The system spaces forward to a point beyond the tape marker that follows the nonstandard labels and positions the tape at the file's data. Each reel must have a tape marker at the end of the file's data. Information beyond this ending tape marker is ignored. Only a single data file can exist on a nonstandard tape. Standard-label volumes *cannot* be processed by using the *NS label processing.

For an *input* file, the end-of-file message is sent to the program using the file when the number of volumes specified in the volume list have been processed, or, if no list of volume identifiers is provided, when the number of reels specified in the REELS parameter are processed.

***BLP:** Standard-label processing is bypassed. Each reel *must* have standard labels. Although each reel is checked for a standard volume label and each file must have at least one standard header label (HDR1) and one standard trailer label (EOV1 or EOF1), most other label information (such as the data file record length or block length) is ignored. The sequence number of each file on the volume is determined only by the number of tape markers between it and the start of tape (in contrast to *SL processing in which the file sequence number stored in the header and trailer labels of each file are used to locate a data file).

Most of the information in the data file trailer label is ignored, but if an end-of-file (EOF) trailer label is found, the end-of-file message is sent to the program using the tape file. If no end-of-file trailer label is encountered by the time the specified number of volumes or reels have been processed (volume identifier list and REELS parameter), the end-of-file message is immediately sent to the program using the tape file. Bypass label processing can be used when the user does not know the name of the file used or when some file label information is incorrect.

***LTM:** The volumes have no labels but do have a single leading tape marker before the first data file. REELS(*LTM) is processed the same as REELS(*NL) except that when SEQNBR(1) is specified for an output file to create the first data file on the tape, a leading tape marker is written at the start of the tape before the first data block.

Element 2: Number of Reels

1: Only one tape reel is processed for the *NL, *LTM, *NS, or *BLP tape file input operation if no list of volume identifiers is provided (VOL parameter).

number-of-reels: Specify the maximum number of reels to be processed for an *NL, *LTM, *NS, or *BLP input tape operation when a list of volume identifiers is not specified (VOL parameter). If the next reel is not on the device when the end of the currently-processing tape is reached, a message is sent to the operator requesting that the next tape be installed on the next tape device. The number-of-reels value is ignored for a standard-label (*SL) file or for any output file.

SEQNBR

Specifies the sequence number of the data file on the tape being processed.

- When standard-label tapes are used, the four-position file sequence number is read from the first header label of the data file.
- When bypass label processing is used or when standard-label tapes are not used, the system counts the tape markers from the start of the tape to locate the correct sequence number data file to be processed.
- When multiple-file, multiple-volume tapes are processed using REELS(*SL), the file sequence numbers continue consecutively through the volumes; thus, each new data file has a sequence number one greater than the previous file, regardless of its volume location.

1: For standard-label tapes (not using bypass label processing), the data file having the sequence number 1 is processed. For nonlabeled tapes and for bypass label processing of standard-label tapes, the first data file on the tape is processed.

***END:** The file is written on the end of the tape. This value is used only for files that are written to tape.

An error message is shown on the display when a tape device file is used to read from a tape and the *END special value is specified in the tape device file.

***NEXT:** The next file in the sequence is processed. This value is used for files read from tape. If the tape is currently in a position that is prior to the first file, the first file on the tape is processed.

An error message is shown on the display when a tape file is used to write to a tape and the *NEXT special value is specified in the tape file.

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

LABEL

Specifies the data file identifier of the data file processed by this tape device file. An identifier is defined only for standard-label tapes and is stored in the header label immediately before the data file.

If a data file identifier is specified for any type of label processing other than *SL, it is ignored.

An identifier is required for a standard label output file, but is optional for an input file because the sequence number uniquely identifies the data file to process.

For an input file or output file with EXTEND(*YES) specified, this parameter specifies the identifier of the data file on the tape. The specified identifier must match the one in the labels of the data file that the SEQNBR parameter specifies; otherwise, an error message is sent to the program using this device file. For output files with EXTEND(*NO) specified, this parameter specifies the identifier of the data file to be created on the tape. More information on this parameter is in Commonly used parameters.

***NONE:** The data file identifier is not specified.

data-file-identifier: Specify the identifier (17 alphanumeric characters maximum) of the data file used with this tape device file. If this identifier is for a tape written in the basic exchange format, and is used on a system other than an iSeries 400, up to eight characters or a qualified identifier having no more than eight characters per qualifier must be used.

FILETYPE

Specifies whether the tape device file being created describes data records or describes source records (statements) for a program or another file. More information on this parameter is in Commonly used parameters.

***DATA:** The tape file describes data records.

***SRC:** The tape file describes source records.

Note: If *SRC is specified, the system adds 12 bytes to the start of every record (to replace the sequence number and date fields).

RCDLEN

Specifies, in bytes, the length of the records contained in the data file processed with this device file. The system always uses the record length and block length specified in the data file labels for any standard-label input file or output file with EXTEND(*YES) specified (if a second header label (HDR2) is found on the tape and *BLP label processing has not been specified).

***CALC:** No record length is specified for the data file being processed. If *CALC is specified, the system will attempt to calculate an appropriate record length when the file is opened. RCDLEN(*CALC) can be used for nonlabeled tapes or when there is no HDR2 label if a BLKLEN value other than *CALC is specified for the file and RCDBLKFMT does not specify spanned or blocked records. In this case, the system calculates an appropriate record length from the block length, record block format, and buffer offset (for an ASCII file) specified for the file. In any other case, the actual record length must be specified by a CHGTAPF command or OVRTAPF command, or in the high-level language program that opens the device file.

record-length: Specify a value ranging from 1 through 32767 bytes that indicates the length of each record in the data file. The minimum and maximum record length allowed for a file is dependent on the record block format, block length, buffer offset (for an ASCII file), and recording code. The *EBCDIC Record Length Values* and *ASCII Record Length Values* tables (at the end of this parameter description) show the minimum and maximum record length values allowed for each record block format, assuming the block length value is large enough to support the maximum record length.

Figure 1. EBCDIC Record Length Vlaues (RCDLEN Parameter)

	EBCDIC RCDLEN Ranges	
RCDFBLKFMT	**FILETYPE(*DATA)**	**FILETYPE(*SRC)**
*F *FB *U	18 - 32767	30 - 32767
*V *VB	1 - 32759	13 - 32767
*VS *VBS	1 - 32759	13 - 32767

	ASCII RCDLEN Ranges	
RCDFBLKFMT	**FILETYPE(*DATA)**	**FILETYPE(*SRC)**
*F *FB *U	18 - 32767	30 - 32767
*D *DB	1 - 9995	13 - 10007
*VS *VBS	1 - 32759	13 - 32767

Figure 2. ASCII Record Length Vlaues (RCDLEN Parameter)

	ASCII RCDLEN Ranges	
RCDFBLKFMT	**FILETYPE(*DATA)**	**FILETYPE(*SRC)**
*F *FB *U	18 - 32767	30 - 32767
*D *DB	1 - 9995	13 - 10007
*VS *VBS	1 - 32759	13 - 32767

BLKLEN

Specifies, in bytes, the maximum length of the data blocks transferred to or from the tape for output or input operations. The system uses the block length and record length specified in the

data file labels for any standard-label input file or output file with EXTEND(*YES) specified (if a second header label (HDR2) is found on the tape and *BLP label processing has not been specified).

***CALC:** No block length is specified for the data file to be processed. If *CALC is specified, the system attempts to calculate an appropriate block length when the file is opened. BLKLEN(*CALC) can be used for nonlabeled tapes or when there is no HDR2 label if a RCDLEN value other than *CALC is specified for the file and RCDBLKFMT does not specify spanned or blocked records. In this case, the system calculates an appropriate block length from the record length, record block format, and buffer offset (for an ASCII file) specified for the file. In any other case, the actual block length must be specified by a CHGTAPF command or OVRTAPF command, or in the high-level language program that opens the device file.

block-length: Specify a value, not exceeding 524288 bytes, that specifies the maximum length of each block in the data file to be processed. The minimum block length that can be successfully processed is determined by the tape device hardware and iSeries 400 machine support functions.

- The maximum block length is always 524288 bytes for an input file, but is limited to 9999 bytes if block descriptors must be created for an ASCII output file.
- The following table shows the minimum and maximum block length values allowed for an output file:

CODE	BUFOFSET	Minimum BLKLEN	Maximum BLKLEN
*EBCDIC	Ignored	18	524288
*ASCII	0	18	524288
*ASCII	*BLKDSC	18	9999

BUFOFSET

Specifies the buffer offset value for the start of the first record in each block in the tape data file. A buffer offset value can be used for any record block format ASCII file, and is ignored for an EBCDIC tape file. The system uses the buffer offset specified in the data file labels for any standard-label input file or output file with EXTEND(*YES) specified if a value is contained in the second header label (HDR2) on the tape, and *BLP label processing has not been specified.

The buffer offset parameter specifies the length of any information that precedes the first record in the block. For record block formats *D, *DB, *VS, and *VBS, each record or record segment is preceded by a descriptor that contains the length of the record or segment. A buffer offset value is used to indicate that there is information *ahead* of the descriptor word for the first record in each block, or *ahead* of the data of the first fixed-length record or undefined format record in each block.

This parameter is not needed for a standard-label file processed for input if the tape includes a second file header label (HDR2) that contains the buffer offset value. A buffer offset value must be provided by the Create Tape File (CRTTAPF) command, Change Tape File (CHGTAPF) command, or Override Tape File (OVRTAPF) command, or by the file labels for an input file that contains any information (such as a block descriptor) ahead of the first record in each block. If the user does not specify a buffer offset value when a tape file is created, it is not necessary to specify an offset value when the file is read.

The only buffer offset values allowed for an output file are zero and *BLKDSC. An existing standard-label data file with a buffer offset value in the HDR2 label can be extended only if the buffer offset value is either 0 or 4. A buffer offset value of 0 in the HDR2 label adds data blocks with *no* buffer offset. BUFOFSET(*BLKDSC) must be specified to extend an existing tape data file that contains an offset value of 4 in the HDR2 label.

0: No buffer offset value precedes the first record in each data block.

***BLKDSC:** Creates 4-byte block descriptors in any tape file created by using this device file. Any input file read by using this device file should assume 4 bytes of buffer offset information

preceding the first record in each data block. This value is valid only for a record block format *D or *DB file. The contents of the buffer offset information of each output data block when BUFOFSET(*BLKDSC) is specified is the actual length of the data block, expressed in zoned decimal format.

buffer-offset: Specify a value ranging from 0 through 99 bytes that specifies the length of the buffer offset information that precedes the first record in each data block.

RCDBLKFMT

Specifies the type and blocking attribute of records in the tape data file being processed.

Record block format *V and *VB records can be processed only for an EBCDIC file; *D and *DB records can be processed only for an ASCII file. If a standard-label tape (label type *SL or *BLP) is being processed and an inconsistent record block format is specified for the volume code, the correct record type is assumed (V or D) for the volume code and a warning message is sent to the program that opens the file. If the record type and code are inconsistent for a nonlabeled volume (label type *NL, *LTM, or *NS), an error message is sent and the file is *not* opened, because there are no labels to verify the correct volume code.

If a valid record length, block length, and buffer offset value (for an ASCII file) are specified for fixed-length records but the block attribute is incorrect, the correct block attribute is assumed (changing record block format *F to *FB or record block format *FB to *F), and a warning message is sent to the program that opens the file.

If a block length is specified that is longer than required to process a maximum length record, then record block format *V, *D, or *VS is changed to *VB, *DB, or *VBS and a warning message is sent to the program that opens the file.

The *Required RCDLEN/BLKLEN/BUFOFSET Relation* table, at the end of this parameter description, shows the required relationship between the record length, block length, and buffer offset (for ASCII) file parameters for an output file or an input file where the file parameters are not determined from a second file header label (HDR2).

Note: When BUFOFSET(*BLKDSC) is specified for the file, a value of 4 should be used for the BUFOFSET part of any BLKLEN calculations, unless existing file labels on the tape specify a different value.

***FB:** Fixed-length, blocked, unspanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *F, based on other file parameters.

***F:** Fixed-length, unblocked, unspanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *FB, based on other file parameters.

***V:** Variable-length, unblocked, unspanned records in EBCDIC type V format are processed. The system may change this record block format to *VB, *D, or *DB, based on other file parameters.

***VB:** Variable-length, blocked, unspanned records in EBCDIC type V format are processed. The system may change this record block format to *DB, based on the volume code.

***D:** Variable-length, unblocked, unspanned records in ASCII type D format are processed. The system may change this record block format to *DB, *V, or *VB, based on other file parameters.

***DB:** Variable-length, blocked, unspanned records in ASCII type D format are processed. The system may change this record block format to *VB, based on the volume code.

***VS:** Variable-length, unblocked, spanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *VBS, based on other file parameters. Note

that the representation of spanned records on the tape is different for EBCDIC and ASCII files, but the system selects the correct format based on the file code.

***VBS:** Variable-length, blocked, spanned records in either EBCDIC or ASCII code are processed. Note that the representation of spanned records on the tape differs for EBCDIC and ASCII files, but the system selects the correct format based on the file code.

***U:** Undefined format records in either EBCDIC or ASCII code are processed. RCDBLKFMT(*U) records are processed as variable-length records, and each record written or read is in a separate tape block. This format can be useful for processing tape files that do not have the formatting requirements of any other record block format.

Figure 3. Required RCDLEN/BLKLEN/BUFOFSET Relation

Table 1. Required RCDLEN/BLKLEN/BUFOFSET Relation

CODE	RCDBLKFMT	BLKLEN[1]
*EBCDIC	*F *U	= RCDLEN
*ASCII	*F *U	= RCDLEN + BUFOFSET
*EBCDIC	*FB	= RCDLEN * n
*ASCII	*FB	= (RCDLEN * n) + BUFOFSET (where n is the number of records in a maximum-length block)
*EBCDIC	*V	= RCDLEN * 8
*ASCII	*D	= RCDLEN * 4 + BUFOFSET
*EBCDIC	*VB	>= RCDLEN + 8
*ASCII	*DB	>= RCDLEN + 4 + BUFOFSET
*EBCDIC	*VS *VBS	>= 18
*ASCII	*BS *VBS	>= 6 + BUFOFSET (18 minimum)

[1] Block length (BLKLEN) is a function of record length (RCDLEN) and buffer offset (BUFOFSET).

EXTEND

Specifies, for output operations to tape, whether new records are added to the end of a data file currently on the tape. The specific data file is identified by the SEQNBR parameter and, for a standard-label file, the LABEL parameter. If the data file is extended, it becomes the last file on the tape volume; data files that follow it are overwritten as the specified file is extended.

Note: This parameter is not valid for 1/4-inch cartridge tape devices.

Element 1: Adding Records to Data File

***NO:** Records are not added to the end of the specified data file. If there is already a data file with the specified SEQNBR on the tape, a new data file is created by overwriting the existing data file and any files that follow it.

***YES:** New records are added to the end of the specified data file on tape when this device file is used.

Element 2: Checking Active Files

***NOCHECK:** The file is extended without being checked to see whether the file is active.

***CHECK:** Before the file is extended, the file is checked to see whether it is active.

DENSITY

Specifies the density of the data that is written on the tape volume when this device file is created. This parameter is used only for tape files being written to tape; it is ignored for tape files being read from the tape (in the case of files being read from tape, the density on the tape is used).

The density of a standard-label volume is specified on the INZTAP command, which initializes tapes as standard-label volumes by writing volume labels on them. If the density specified on this parameter is different than the density of a standard-labeled tape, the density currently on tape is used and a warning message is sent. The density of a standard-label volume can only be changed by re-initializing the tape.

***DEVTYPE:** The highest capacity density or format supported by the tape device will be used.

Tape device	Highest capacity density or format
2440	6250
3422	6250
3430	6250
3480	*FMT3480
3490E	*FMT3490E
3570-BXX	*FMT3570
3570-CXX	*FMT3570E
3580-001	*ULTRIUM1
3590	*FMT3590
3590-Exx	*FMT3590E
6335	*QIC3040
6341	*QIC120
6342	*QIC525
6343	*QIC1000
6344	*QIC2GB
6346	*QIC120
6347	*QIC525
6348	*QIC1000
6349	*QIC2GB
6366	*QIC120
6368	*QIC1000
6369	*QIC2GB
6378	*QIC525
6379	*QIC1000
6380	*QIC2GB

6381 *QIC2DC

6382 *QIC4DC

6383 *QIC5010

6385 *QIC5010

6386 *MLR3

6387 *SLR100

6390 *FMT7GB

7207-122
*QIC4DC

7208-002
*FMT2GB

7208-012
*FMT5GB

7208-222
*FMT7GB

7208-342
*FMT20GB

9346 *QIC120

9347 3200

9348 6250

***CTGTYPE:** The highest capacity density or format supported by the device for the mounted cartridge type will be used. If the device does not support special cartridge type information, *DEVTYPE is used.

tape-density: Specify the density or format to use.

1600 The data density on the tape volume is 1,600 bits per inch, which is used for 1/2 inch reel tapes.

3200 The data density on the tape volume is 3,200 bits per inch, which is used for 1/2 inch reel tapes.

6250 The data density on the tape volume is 6,250 bits per inch, which is used for 1/2 inch reel tapes.

***FMT3480**
The format of this tape is FMT3480. The data density on this tape volume is formatted to support a 3480 device. This density is used for 1/2 inch cartridge tapes.

***FMT3490E**
The format of this tape is FMT3490E. The data density on this tape volume is formatted to support a 3490E device. This density is used for 1/2 inch cartridge tapes.

***FMT3570**
The format of this tape is FMT3570. The data format is written on the tape volume with a 3570 device.

***FMT3570E**
The format of this tape is FMT3570E. The data format is written on the tape volume with a 3570E device.

***FMT3590**
The format of this tape is FMT3590. The data format is written on the tape volume with a 3590 device. This density is used for 1/2 inch cartridge tapes.

***FMT3590E**
The format of this tape is FMT3590E. The data format is written on the tape volume with a 3590E device. This density is used for 1/2 inch cartridge tapes.

***QIC120**
The format of this tape is QIC120, which is used for 1/4 inch cartridge tapes that can hold 120 megabytes of data.

***QIC525**
The format of this tape is QIC525, which is used for 1/4 inch cartridge tapes that can hold 525 megabytes of data.

***QIC1000**
The format of this tape is QIC1000, which is used for 1/4 inch cartridge tapes that can hold 1200 megabytes of data.

***QIC2GB**
The format of this tape is QIC2GB. It is used by 1/4 inch tape devices which can store 2.5 gigabytes of data on a standard length QIC2GB cartridge.

***QIC2DC**
The format of this tape is QIC2DC. It is used to write compacted data to a 1/4 inch cartridge that supports the QIC2GB format.

***QIC4GB**
The format of this tape is QIC4GB. It is used by 1/4 inch tape devices which can store 4 gigabytes of data on a standard length QIC4GB cartridge.

***QIC4DC**
The format of this tape is QIC4DC. It is used to write compacted data to a 1/4 inch cartridge that supports the QIC4GB format.

***QIC3040**
The format of this tape is QIC3040, which is used for 1/4 inch minicartridge tapes that can hold 840 megabytes of data.

***QIC5010**
The format of this tape is QIC5010, which is used for 1/4 inch cartridge tapes that can hold 13.5 gigabytes of data.

***MLR3**
The format of this tape is MLR3. It is used by 1/4 inch tape devices which can store 25 gigabytes of data on a standard length MLR3 cartridge.

***SLR100**
The format of this tape is SLR100. It is used by 1/4 inch tape devices which can typically store 100 gigabytes of compacted data on a standard length SLR100 cartridge.

***FMT2GB**
The format of this tape is FMT2GB, which is used for 8 millimeter cartridge tapes that can hold 2 gigabytes of data.

***FMT5GB**
The format of this tape is FMT5GB, which is used for 8 millimeter cartridge tapes that can hold 5 gigabytes of data.

***FMT7GB**
The format of this tape is FMT7GB, which is used for 8 millimeter cartridge tapes that can hold 7 gigabytes of data.

***FMT20GB**
The format of this tape is FMT20GB. It is used by 8 millimeter tape devices that can store 20 gigabytes of data on a standard length cartridge.

***ULTRIUM1**
The format of this tape is ULTRIUM1. It is used by 1/2 inch cartridge tape devices that can store 100 gigabytes of data on a standard length cartridge.

Note: Some of the density values shown can only be specified when a tape device which supports that density is attached to the system.

Note: Self-configured tape devices may define additional valid values for the density parameter. Use iSeries 400 Operations Navigator (Configuration and Service) (Hardware) (Tape Units) (Properties) to find additional valid density values for a specific device, or use the F4=Prompt key on the Tape density field of the CL command to see a list of all valid density values for the attached tape devices.

COMPACT
Specifies whether device data compaction is performed. If the tape devices being used do not support data compaction, this parameter will be ignored when the file is opened.

***DEVD:** Device data compaction is performed if the devices being used support data compaction.

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

CODE Specifies the character code used. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

CRTDATE
Specifies, for tape input data files and for tape output for which EXTEND(*YES) is specified, the date when the data file was created (written on tape).

Note: The data file creation date is stored in file labels on the tape. If a creation date is specified for any type of label processing other than standard-label (*SL), it is ignored. If the creation date written on the tape containing the data file does not match the date specified in this device file description, an inquiry message is sent to the operator.

***NONE:** The creation date is not specified. It is not checked unless it is supplied before the device file is opened, either in a OVRTAPF command or CHGTAPF command, or in the high-level language program.

creation-date: Specify the creation date of the data file used by this device file. The date must be specified in the format defined by the job attributes DATFMT and, if separators are used, DATSEP.

EXPDATE

Specifies, for tape output data files only, and only when standard-labeled tapes are used, the expiration date of the data file used by this device file. If a date is specified, the data file is protected and cannot be overwritten until after the specified expiration date. The files cannot be overwritten until after the expiration date.

Note: The data file expiration date is stored in file labels on the tape. If an expiration date is specified for any type of label processing other than *SL, it is ignored.

***NONE:** No expiration date for the data file is specified; the file is not protected. An expiration date is written in the data file labels so the file can be used as a scratch data file.

***PERM:** The data file is permanently protected. An expiration date of 999999 is assigned.

expiration-date: Specify the date on which the data file expires, after which it can be overwritten with new data. The expiration date must be later than or equal to the current date. The date must be specified in the format defined by the job attributes QDATFMT and, if separators are used, QDATSEP.

ENDOPT

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

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

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

This option is used to reduce the time required to position the tape if the next tape file that opens to this device uses a data file that is on this volume.

Note: Even if ENDOPT(*LEAVE) is specified, the next tape file opened to this reel is positioned at the beginning of some data file on the volume (or at the end of a data file, for either read backward or for output that extends an existing data file on the volume). A tape file is always positioned at the start or end of a data file when it is opened.

***UNLOAD:** The tape is automatically rewound and unloaded after the operation ends.

USRLBLPGM

Specifies the qualified name of the user program that processes user tape labels. On an output file, the user label program will pass the user labels that are written to tape. On an input file, the user labels are passed to the user label program.

***NONE:** There is no user label program for this device file.

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

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

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

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

user-label-program-name: Specify the name of the user program that processes the user tape labels. If no library qualifier is given, *LIBL is used to find the file.

IGCDTA

Specifies whether the file processes double-byte character set (DBCS) data.

***NO:** The file does not process DBCS data.

***YES:** The file processes DBCS data.

WAITFILE

Specifies the number of seconds that the program waits for the file resources and session resources to be allocated when the file is opened, or for the device or session resources to be allocated when an acquire operation is performed to the file. If those resources are not allocated within the specified wait time, an error message is sent to the program. More information on this parameter is in Commonly used parameters.

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

***IMMED:** The program does not wait; when the file is opened, an immediate allocation of the file resources is required.

***CLS:** The job default wait time is used as the wait time for the file resources being allocated.

number-of-seconds: Specify the number of seconds that the program waits for the file resources to be allocated to the tape when the file is opened, or the wait time for the device allocated when an acquire operation is performed to the file. Valid values range from 1 through 32767 seconds.

SHARE

Specifies whether the open data path (ODP) for the tape file is shared with other programs in the routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the Database Programming topic in the Information Center.

***NO:** The ODP created by the program with this attribute is not shared with other programs in the routing step. Every time a program opens the file with this attribute, a new ODP to the file is created and activated.

***YES:** The ODP created with this attribute is shared with each program in the routing step that also specifies SHARE(*YES) when it opens the file.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

AUT Specifies the authority given to users who do not have specific authority to the tape file, who are not on an authorization list, and whose user group has no specific authority to the tape file.

***LIBCRTAUT:** The public authority for the tape file is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the tape file). The public authority is determined when the tape file is created. If the CRTAUT value for the library changes after the tape file is created, the new value does not affect any existing objects.

***CHANGE:** The user can perform all operations on the tape file except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the tape file. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the tape file.

***USE:** The user can perform basic operations on the tape file, such as running a program or reading a file. The user cannot change the tape file. *USE authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The user cannot access the tape file.

authorization-list-name: Specify the name of the authorization list used.

REPLACE

Specifies whether the existing tape file is replaced by the new file. More information on this parameter is in Commonly used parameters.

***YES:** The existing tape file is replaced by the one being created.

***NO:** The existing file, if any, is not replaced by the tape file.

TEXT Specifies the text that briefly describes the tape file. More information on this parameter is in Commonly used parameters.

***BLANK:** Text is not specified.

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

Examples for CRTTAPF

Example 1: Creating a Description of a Tape Device File

```
CRTTAPF   FILE(BACKHST)  DEV(QTAPE1 QTAPE2 QTAPE3)
  REELS(*BLP 10)  RCDLEN(256)  BLKLEN(1024)
  RCDBLKFMT(*FB)  EXTEND(*YES)  ENDOPT(*UNLOAD)
  WAITFILE(60)
```

This command creates a description of the tape device file named BACKHST in the current library, to be used with the tape devices QTAPE1, QTAPE2, and QTAPE3. All volumes processed on these devices with this device file must have standard-labels. Each block of data (EBCDIC character code) on the tape volumes contains four records of 256 bytes each. When records are written to the tape, they are added to the end of the data file. No creation or expiration date is specified for this tape, and both unloading and rewinding operations will occur when the device file is closed at the last tape volume processed. The program using this tape device file waits 60 seconds for file resources to be allocated when this file is opened, and this device file is used only when the current program is running.

Example 2: Creating a Tape File Containing DBCS Data

```
CRTTAPF   FILE(IGCLIB/IGCTAP)  LABEL(GENINF)
  IGCDTA(*YES)
```

This command creates the tape file IGCTAP that is stored in library IGCLIB, which is labeled GENINF, and contains DBCS data.

Error messages for CRTTAPF

***ESCAPE Messages**

CPF7302
File &1 not created in library &2.

CRTUSRPRF (Create User Profile) Command Description

CRTUSRPRF Command syntax diagram

Purpose

The Create User Profile (CRTUSRPRF) command identifies a user to the system and allows you to customize the way the system appears. When the profile is created, the profile is given *CHANGE and *OBJMGT authorities for the profile itself. The system relies on the profile having these authorities to itself and they should not be removed.

Restriction: The user of this command must have (1) *SECADM special authority, (2) *USE authority to the initial program, initial menu, job description, message queue, output queue, and attention-key-handling program if specified, and (3) *CHANGE and object management authorities to the group profile and supplemental group profiles, if specified.

Required Parameter

USRPRF
Specifies the user name for this user profile. The character set for user profile names is the same set used for *SNAME type on the parameter statement of command definitions source statements. A numeric user profile can be specified. If the user profile is numeric, then it must begin with a *Q*.

Optional Parameters

PASSWORD
Specifies the password that lets the user sign on the system. The password is associated with a unique user name used to represent the user in the system. The password should be known only to the user. A numeric password can be specified.

When the system is operating at password level 0 or 1 and the password is numeric, then it must begin with a *Q*, for example, *Q1234* where *1234* is the password used for signing on the system.

Note: The password level is controlled by the Password Level (QPWDLVL) system value.

Note: The new password is not checked against the password validation rules. The password validation rules are defined by OS/400 system values. For a description of the password validation rules, see the iSeries Security Reference  book.

***USRPRF:** The password for this user is the user profile name specified in the USRPRF parameter. When the system is operating at password level 2 or 3 and the *USRPRF value was specified for the user profile password, the user must enter their password using upper case characters.

***NONE:** No password is used by this user. Users cannot sign on a system with a profile that has PASSWORD(*NONE) specified.

'user-password': When the system is operating at password level 0 or 1, specify an alphanumeric character string (10 characters or less). The first character must be alphabetic and the other characters must be alphanumeric.

When the system is operating at password level 2 or 3, specify a character string (128 characters or less).

SPCAUT

Specifies the special authorities granted to a user. Special authorities are *required* to perform certain functions on the system. Special authorities cannot be removed from many of the system supplied user profiles including QSECOFR and QSYS.

Restrictions:

1. The user profile that is creating or changing another user profile must have all the special authorities being granted.
2. A user must have *ALLOBJ and *SECADM special authorities to grant another user *SECADM special authority when using the CRTUSRPRF command.
3. The user must have *ALLOBJ, *SECADM, and *AUDIT special authorities to grant another user *AUDIT special authority when using the CRTUSRPRF command.

***USRCLS:** Special authorities are granted to the user based on the value specified in the USRCLS parameter.

***NONE:** No special authorities are granted to the user.

***ALLOBJ:** All object authority is granted to the user. It is granted to users who work with system resources. The user can access any system resource whether or not the user has private authorizations.

***AUDIT:** Audit authority is granted to the user. This user is given the authority to perform auditing functions. Auditing functions include turning auditing on or off for the system and controlling the level of auditing on an object or user.

***IOSYSCFG:** Input/Output system configuration authority is granted to the user. The user has authority to change system I/O configurations.

***JOBCTL:** Job control authority is granted to the user. It is normally granted to users who operate the system. The user has authority to change, display, hold, release, cancel, and clear all jobs running on the system or on a job queue or output queue for which OPRCTL (*YES) is specified. The user also has the authority to start writers, and to stop active subsystems.

***SAVSYS:** Save system authority is granted to this user profile. It is normally granted to users who operate the system. The user has authority to save, restore, and free storage for all objects on the system, whether or not object management authority has been granted.

***SECADM:** Security administrator authority is granted to the user. It is granted to users who are the security administrators. The user can create, change, or delete user profiles if authorized to the Create User Profile, Change User Profile, or Delete User Profile commands, and is authorized to the user profile. This authority does not allow the user to grant special authorities that this user profile does not have.

***SERVICE:** Service authority is granted to the user. The user can perform service functions.

***SPLCTL:** Spool control authority is granted to the user. The user can perform all spool functions.

PWDEXP

Specifies whether the user's password is set to expired. If the password is set to expired, the user is required to change the password to sign on the system. When the user attempts to sign on the system, the sign-on information display is shown and the user has the option to change this password.

***NO:** The password is not set to expired.

***YES:** The password is set to expired.

STATUS

Specifies profile status.

***ENABLED:** The profile created is valid for sign-on.

***DISABLED:** The profile created is not valid for sign-on until an authorized user enables it again.

USRCLS

Specifies the class of user associated with this user profile: security officer, security administrator, programmer, system operator, or user. User class determines which menu options are shown. The special authorities defined by the user class are used only if SPCAUT(*USRCLS) is specified. If SPCAUT(*USRCLS) is specified, the special authorities granted will differ depending on the QSECURITY value.

***SECOFR:** At all levels of security, the security officer has the following special authorities:

*ALLOBJ

*SAVSYS

*JOBCTL

*SERVICE

*SPLCTL

*SECADM

*AUDIT

*IOSYSCFG

***SECADM:** At QSECURITY level 10 or 20, the security administrator has *ALLOBJ, *SAVSYS, *SECADM, and *JOBCTL special authorities.

At QSECURITY level 30 or above, the user has *SECADM special authority.

***PGMR:** At QSECURITY level 10 or 20, the programmer has *ALLOBJ, *SAVSYS, and *JOBCTL special authorities.

At QSECURITY level 30 or above, the user has no special authorities.

***SYSOPR:** At QSECURITY level 10 or 20, the system operator has *ALLOBJ, *SAVSYS, and JOBCTL special authorities.

At QSECURITY level 30 or above, the user has *SAVSYS and *JOBCTL special authorities.

***USER:** At QSECURITY level 10 or 20, the user is granted *ALLOBJ and *SAVSYS authority.

At QSECURITY level 30 or above, this user is granted no special authorities.

ASTLVL

Specifies which user interface to use.

***SYSVAL:** The assistance level defined in the system value, QASTLVL, is used.

***BASIC:** The Operational Assistant* user interface is used.

***INTERMED:** The system interface is used.

***ADVANCED:** The expert system interface is used. To allow for more list entries, the options keys and the function keys are not displayed. If a command does not have an advanced (*ADVANCED) level, the intermediate (*INTERMED) level is used.

SPCENV

Specifies the special environment in which the user operates after signing on the system.

***SYSVAL:** The system value QSPCENV is used to determine the system environment after the user signs on the system.

***NONE:** The user operates in the iSeries 400 environment after signing on the system.

***S36:** The user operates in the System/36 environment after signing on the system.

DSPSGNINF

Specifies whether the sign-on information is displayed when the user signs on. This allows users to see the sign-on information, such as date of last sign-on and number of sign-on attempts that were not valid. If the password is due to expire in 7 days or less, the number of days until the password expires is shown.

***SYSVAL:** The system value QDSPSGNINF is used to determine whether the sign-on information display is shown when the user signs on the system.

***NO:** The sign-on information display is not shown when the user signs on the system.

***YES:** The sign-on information display is shown when the user signs on the system.

PWDEXPITV

Specifies the interval (number of days from the password changed date) before the password expires.

***SYSVAL:** The system value QPWDEXPITV is used to determine the password expiration interval.

***NOMAX:** The password does not expire.

password-expiration-interval: Specify the number of days between when the password was last changed and the date when the password expires. Valid values range from 1 through 366.

LMTDEVSSN

Specifies whether the user is limited to one device session. This does not limit use of the System Request menu or prevent a second sign-on.

***SYSVAL:** The system value QLMTDEVSSN determines whether the user is limited to one device session.

***NO:** The user is not limited to one device session.

***YES:** The user is limited to one device session.

KBDBUF

Specifies the keyboard buffering value used when a job is initialized for this user profile. If the type-ahead feature is active, the keystrokes can be buffered. If the attention key buffering option is active, the attention key is buffered like any other key. If the attention key is not active, the attention key is not buffered and is sent to the system even if the display station is input inhibited. The keyboard buffer value can also be set by a user application using the QWSSETWS program.

***SYSVAL:** The system value, QKBDBUF, is used to determine the keyboard buffering value for this profile.

***NO:** The type-ahead feature and attention key buffering option are not active for this user profile.

***TYPEAHEAD:** The type-ahead feature is active for this user profile.

***YES:** The type-ahead feature and attention key buffering option are active for this user profile.

MAXSTG

Specifies the maximum auxiliary storage space assigned to store permanent objects owned by this

user profile. If the storage maximum is exceeded when an interactive user creates an object, an error message is displayed and the object is not created. If the maximum is exceeded when an object is created in a batch job, an error message is sent to the job log (depending on the logging level of the job) and the object is not created.

Storage is allocated in 4K increments. Therefore, if you specify MAXSTG (9), the profile is allocated 12K of storage.

When planning maximum storage space for user profiles, consider the following system actions:

- A restore operation first assigns storage space to the user doing the restore operation, and then transfers the object to the owner. For a large restore operation, specify MAXSTG(*NOMAX).
- The user profile that creates a journal receiver is assigned storage space as the receiver size grows. If new receivers are created using JRNRCV(*GEN), the storage continues to be assigned to the user profile that owns the active journal receiver. For a very active journal receiver, specify MAXSTG(*NOMAX).
- User profiles that transfer all created objects to their group profile must have adequate storage in the user profiles to contain any created object before it is transferred to the group profile.
- The owner of the library is assigned storage space for the descriptions of objects stored in a library, even when those objects are owned by another profile. Examples of such objects are text and program references.
- Storage space is assigned to the user profile for temporary objects used when the job is running. Examples of such objects are commit control blocks, file editing space, and documents.

***NOMAX:** As much storage as is required is assigned to this profile.

maximum-K-bytes: Specify the maximum amount of storage in kilobytes (1KB equals 1024 bytes) that can be assigned to this user profile.

PTYLMT

Specifies the highest scheduling priority the user is allowed to have for each job submitted to the system. The value specified for this parameter controls the job processing priority and output priority for any job running under this user profile. This means that values specified in the JOBPTY and OUTPTY parameters of any job command cannot exceed the PTYLMT value of the user profile under which the job is running. The scheduling priority value ranges from 0 through 9, where 0 is the highest priority and 9 is the lowest priority.

3: The user named in this profile can have a priority value no higher than 3 for scheduling jobs on the system. Jobs with this priority value are run before jobs with priority values ranging from 4 through 9, and after jobs with priority values ranging from 0 through 2.

priority-limit: Specify a value, ranging from 0 through 9, for the user's highest job scheduling priority.

CURLIB

Specifies the name of the library being used as the current library for jobs initiated by this user profile. If *PARTIAL or *YES is specified for the LMTCPB parameter in the user profile, the user cannot change the current library at sign-on or with the Change Profile (CHGPRF) command.

***CRTDFT:** The user has no current job library. If objects are created in the current library by using *CURLIB on a create command, the QGPL library is used as the default current library.

current-library-name: Specify the 10-character name of the user's current library after the user signs on the system.

INLPGM

Specifies, for an interactive job, the name of the program called whenever a new routing step is started that has QCMD as the request processing program. No parameters can be passed to the program. If *PARTIAL or *YES is specified on the LMTCPB parameter in the user profile, the initial program value cannot be changed at sign-on or with the Change Profile (CHGPRF) command.

A System/36 environment procedure name can be specified as the initial program if the procedure is a member of the file QS36PRC (in the library list or specified library) and if either of the following conditions are true:

- *S36 is specified for the SPCENV parameter.
- *SYSVAL is specified for the SPCENV parameter and the system value, QSPCENV, is *S36.

***NONE:** No program is called when the user signs on the system. If a menu name is specified in the INLMNU parameter, that menu is displayed.

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

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

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

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

program-name: Specify the name of the program called when the user signs on the system.

INLMNU

Specifies the name of the menu displayed when the user signs on the system if the user's routing program is the command processor QCMD. If *YES is specified on the LMTCPB parameter in the user profile, the user cannot change this menu.

A System/36 environment menu can be specified as the initial menu if either of the following conditions are true:

- *S36 is specified for the SPCENV parameter.
- *SYSVAL is specified for the SPCENV parameter and the system value, QSPCENV, is *S36.

***SIGNOFF:** The system signs off the user when the initial program completes. This is intended for users limited to only running the initial program.

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

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

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

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

menu-name: Specify the name of the menu shown when the user signs on the system.

LMTCPB

Specifies the limits of user control over the initial program, the initial menu, the current library, and the ATTN key handling program values. This parameter is ignored when the security level is 10.

Note: When creating or changing other users' user profiles, users running this command cannot specify values on this parameter that grant greater capabilities to other users than their own user profiles grant to them. For example, if *PARTIAL is specified on the LMTCPB parameter in the user profile of the user running this command, *PARTIAL or *YES can be specified for another user. *NO cannot be specified for another user.

***NO:** The program, menu, and current library values can be changed when the user signs on the system. A user can change the initial program, the initial menu, the current library, or the ATTN key handling program values in the user's user profile by using the Change Profile (CHGPRF) command.

***PARTIAL:** The initial program and the current library cannot be changed on the Sign-On display. The initial menu can be changed and commands can be run from any command line. A user can change the initial menu value by using the Change Profile (CHGPRF) command. The initial program, the current library, and the ATTN key handling program cannot be changed by using the Change Profile command.

***YES:** The program, the menu, and the current library values cannot be changed on the sign-on display. Commands cannot be run when issued from a command line or by selecting an option from a command grouping menu such as CMDADD, but can still be run from a command entry screen. The user cannot change the initial program, the initial menu, the current library, or the ATTN key handling program by using the Change Profile (CHGPRF) command.

JOBD Specifies the name of the job description used for jobs that start through subsystem work station entries. If the job description does not exist when the user profile is created or changed, a library qualifier must be specified, because the job description name is kept in the user profile.

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

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

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

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

QDFTJOBD: The default system-supplied job description found in the QGPL library is used.

job-description-name: Specify the name of the job description that is used for the work station entries whose job description parameter values indicate the user (JOBD(*USRPRF)).

GRPPRF

Specifies the name of the group profile whose authority is used if the user has no specific authority for the job. The current user of this command must have *OBJMGT, *CHANGE, *OBJOPR, *READ, *ADD, *UPD, and *DLT authorities to the profile specified on the GRPPRF parameter. The required *OBJMGT cannot come from the program adopt operation.

Notes:

1. When a group profile is specified, the user is automatically granted *CHANGE, *OBJMGT, *OBJOPR, *READ, *ADD, *UPD, AND *DLT authority to the group profile.

2. The following IBM-supplied objects are not valid on this parameter:

```
QAUTPROF        QFNC            QSNADS
QCLUMGT         QGATE           QSPL
QCLUSTER        QIPP            QSPLJOB
QCOLSRV         QLPAUTO         QSRV
QDBSHR          QLPINSTALL      QSRVBAS
QDBSHRDO        QMSF            QSYS
QDFTOWN         QNETSPLF        QTCM
QDIRSRV         QNFSANON        QTCP
QDLFM           QNTP            QTFTP
QDOC            QPEX            QTSTRQS
QDSNX           QPM400          QYPSJSVR
QEJB            QRJE
```

***NONE:** This user profile has no group profile.

user-profile-name: Specify the name of a group profile used with this user profile.

OWNER

Specifies whether the user profile or the group profile is the owner of newly created objects.

***USRPRF:** The user profile being used with the job is made the owner of the newly created object.

***GRPPRF:** The group profile is made, the owner of newly created objects and has all authority to those objects. The user profile used with the job does not have specific authority to the object. If *GRPPRF is specified, there must be a user profile name in the GRPPRF parameter, and the GRPAUT parameter cannot be specified.

GRPAUT

Specifies the specific group authority granted to the group profile for newly created objects. If *GRPPRF is specified on the OWNER parameter, specification of this parameter is not allowed.

***NONE:** No group authority is granted.

***ALL:** The group profile is granted authority to perform all operations on the object except those limited to the owner or controlled by authorization list management authority. In addition to the functions allowed with use (*USE) and change (*CHANGE) authorities, the group profile can control the object's existence and specify the security for the object. The group profile can transfer ownership of the object. If the object is an authorization list, then the group profile cannot add, change, or remove users.

***CHANGE:** The group profile is granted authority to perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. In addition to the functions allowed with use (*USE) authority, the group profile can change the object. Change authority provides object operational authority and all of the data authorities.

***USE:** The group profile runs a program or displays the contents of a file. The group profile is prevented from changing the object. Use authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The group profile is prevented from accessing the object.

GRPAUTTYP

Specifies the type of authority to be granted to the group profile for newly-created objects. If the authority value in the GRPAUT parameter is *NONE, this value is ignored.

***PRIVATE:** The group profile is granted private authority to newly-created objects, with the authority value determined by the GRPAUT parameter.

***PGP:** The group profile is the primary group of newly-created objects, with the authority value determined by the GRPAUT parameter.

SUPGRPPRF

Specifies the user's supplemental group profiles. The profiles specified here, along with the group profile specified on the GRPPRF parameter, are used to determine what authority the user has if no specific user authority is given for the job. If profiles are specified for this parameter, a group profile name must be specified on the GRPPRF parameter. The current user of this command must have *OBJMGT, *OBJOPR, *READ, *ADD, *UPD, and *DLT authority to the profiles specified on the SUPGRPPRF parameter. The required *OBJMGT authority cannot be given by a program adopt operation.

Notes:

1. When a group profile is specified, the user is automatically granted *OBJMGT, *OBJOPR, *READ, *ADD, *UPD, AND *DLT authority to the group profile.
2. The following IBM-supplied objects are not valid on this parameter:

```
QAUTPROF        QFNC            QSNADS
QCLUMGT         QGATE           QSPL
QCLUSTER        QIPP            QSPLJOB
QCOLSRV         QLPAUTO         QSRV
QDBSHR          QLPINSTALL      QSRVBAS
QDBSHRDO        QMSF            QSYS
QDFTOWN         QNETSPLF        QTCM
QDIRSRV         QNFSANON        QTCP
QDLFM           QNTP            QTFTP
QDOC            QPEX            QTSTRQS
QDSNX           QPM400          QYPSJSVR
QEJB            QRJE
```

***NONE:** No supplemental group profiles are used with this user profile.

group-profile-name: Specify a maximum of 15 group profile names used with this user profile and the group profile specified on the GRPPRF parameter to determine a job's eligibility for getting access to existing objects and special authority.

ACGCDE

Specifies the accounting code associated with this user profile. More information on job accounting is in the Database Programming topic in the Information Center.

***BLANK:** A 15-blank job accounting code is assigned to this user profile.

job-accounting-code: Specify the 15-character job accounting code used by jobs that get their accounting code from this user profile. If less than 15 characters are specified, the string is padded on the right with blanks.

DOCPWD

Specifies the document password that allows Document Interchange Architecture (DIA) users to protect personal distributions from being used by others working on their behalf. More information on specifying a document password is in the Managing OfficeVision/400 book.

***NONE:** The user has no document password.

document-password: Specify a document password for use by this user. The password must range from 1 through 8 alphanumeric characters (letters A through Z and numbers 0 through 9). The first character of the document password must be alphabetic; the remaining characters can be alphanumeric. Embedded blanks, leading blanks, and special characters are invalid.

MSGQ

Specifies the name of the message queue used by this user. The message queue is created if it does not already exist. The user profile being created is the owner of the message queue.

***USRPRF:** A message queue with the same name as that specified in the USRPRF parameter is the message queue for this user. This message queue is located in QUSRSYS.

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

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

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

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

message-queue-name: Specify the name of the qualified message queue for this user.

DLVRY

Specifies how the messages that are sent to the message queue for the user profile are delivered.

***NOTIFY:** The job to which the message queue is assigned is notified when a message arrives at the message queue. For interactive jobs at a work station, the audible alarm is sounded and the message-waiting light is turned on. The delivery mode cannot be changed to *NOTIFY if the message queue is also being used by another job.

***HOLD:** The messages are held in the message queue until they are requested by the user or program.

***BREAK:** The job to which the message queue is assigned is interrupted when a message arrives at the message queue. If the job is an interactive job, the audible alarm (if present on the display station) is sounded. The delivery mode cannot be changed to *BREAK if the message queue is also being used by another job.

***DFT:** The default reply to the inquiry message is sent. If no default reply is specified in the message description of the inquiry message, the system default reply, *N, is used.

SEV Specifies the lowest severity code that a message can have and still be delivered to a user in break or notify mode. Messages arriving at the message queue whose severities are lower than the severity code specified on this parameter do not interrupt the job or turn on the audible alarm or the message-waiting light; they are held in the queue until they are requested by using the Display Message (DSPMSG) command. If *BREAK or *NOTIFY is specified on the DLVRY parameter, and is in effect when a message arrives at the queue, the message is delivered if the severity code associated with the message is equal to or greater than the value specified here. Otherwise, the message is held in the queue until it is requested.

00: The default severity level is 00.

severity-code: Specify a value, ranging from 00 through 99, for the lowest severity code that a message can have and still be delivered if the message queue is in break or notify delivery mode.

PRTDEV

Specifies the name of the default printer device for this user. If the printer file being used to create the output specifies to spool the file, the spooled file is placed on the device's output queue, which is named the same as the device.

Note: This assumes the defaults are specified on the OUTQ parameter for the printer file, job description, user profile and workstation.

***WRKSTN:** The output queue assigned to the user's work station is used.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

print-device-name: Specify the name of the printer used to print the output for this user.

OUTQ Specifies the qualified name of the output queue.

***WRKSTN:** The output queue assigned to the user's work station is used.

***DEV:** The output queue associated with the printer specified on the DEV parameter is used. The output queue has the same name as the printer. (The printer file DEV parameter is determined by the Create Printer File (CRTPRTF), Change Printer File (CHGPRTF), or Override with Printer File (OVRPRTF) command.

Note: This assumes the defaults were specified on the OUTQ parameter for the printer file, job description, user profile, and workstation.

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

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

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

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

output-queue-name: Specify the name of the output queue for this user.

ATNPGM

Specifies the program used for the ATTN key handling program for this user. The ATTN key handling program is called when the ATTN key is pressed in an interactive job. The program is active only when the user routes requests to the QCMD command processor supplied with the system. The ATTN key handling program is set on before a program (if any) is called and is active for both a program and a menu. If a program changes the ATNPGM by using the Set Attention Program (SETATNPGM) command, the new program remains active only for the duration of the program. When control returns and QCMD calls the menu, the original ATTN key handling program again becomes active. If the SETATNPGM command is run from a menu or an application is called from a menu, the new ATTN key handling program specified overrides the original ATTN key handling program. If *YES or *PARTIAL is specified on the LMTCPB parameter in the user profile, the ATTN key handling program cannot be changed with the Change Profile (CHGPRF) command.

***SYSVAL:** The system value QATNPGM is used.

***ASSIST:** QEZMAIN is used.

***NONE:** No ATTN key handling program is used by this user.

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

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

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

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

program-name: Specify the name of the ATTN key handling program used by this user.

SRTSEQ

Specifies the sort sequence table to be used for string comparisons for this user profile.

***SYSVAL:** The system value QSRTSEQ is used.

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

***LANGIDUNQ:** A unique-weight sort table is used.

***LANGIDSHR:** A shared-weight sort table is used.

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

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

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

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

table-name: Specify the name of the sort sequence table to be used with this user profile.

LANGID

Specifies the language identifier used for this user.

***SYSVAL:** The system value QLANGID is used.

language-identifier: Specify the language identifier. More information on valid language identifiers is in the Globalization topic in the Information Center.

CNTRYID

Specifies the country or region identifier used for this user.

***SYSVAL:** The system value QCNTRYID is used.

country-identifier: Specify an ISO 3166 Alpha-2 code from the country or region code table. More information on this parameter is in Commonly used parameters.

CCSID

Specifies the coded character set identifier (CCSID) used for this user.

A CCSID is a 16-bit number identifying a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and additional coding-related information that uniquely identifies the coded graphic representation used.

***SYSVAL:** The system value QCCSID is used.

***HEX:** The CCSID 65535 is used.

coded-character-set-identifier: Specify the CCSID. More information on valid CCSIDs is in the Globalization topic in the Information Center.

CHRIDCTL

Sepcifies the character identifier control for the job. This attribute controls the type of CCSID conversion that occurs for display files, printer files and panel groups. The *CHRIDCTL special value must be specified on the CHRID command parameter on the create, change or override commands for display files, printer files and panel groups before this attribute will be used.

***SYSVAL:** The value in the QCHRIDCTL system value will be used.

***DEVD:** The *DEVD special value performs the same function as on the CHRID command parameter for display files, printer files and panel groups.

***JOBCCSID:** The *JOBCCSID special value performs the same function as on the CHRID command parameter for display files, printer files and panel groups.

SETJOBATR

Specifies which job attributes are taken from the locale specified on the LOCALE parameter when the job is initiated.

***SYSVAL:** The system value QSETJOBATR is used to determine which job attributes are taken from the locale.

***NONE:** No job attributes are taken from the locale.

Any combination of the following values can be specified:

***CCSID:** The coded character set identifier from the locale is used. The CCSID value from the locale overrides the user profile CCSID.

***DATFMT:** The date format from the locale is used.

***DATSEP:** The date separator from the locale is used.

***DECFMT:** The decimal format from the locale is used.

***SRTSEQ:** The sort sequence from the locale is used. The sort sequence from the locale overrides the user profile sort sequence.

***TIMSEP:** The time separator from the locale is used.

USROPT

Specifies the level of detail the user sees and the default function of the Page Up and Page Down keys. The system shows displays suitable for the inexperienced user. More experienced users must perform an extra action to see more detailed information. When values are specified for this parameter, the system presents detailed information without additional action by the experienced user.

***NONE:** No detailed information is shown.

***CLKWD:** Parameter keywords are shown instead of the possible parameter values when a command is displayed.

***EXPERT:** More detailed information is initially shown when the user is performing display and edit options such as the edit or display object authority.

***ROLLKEY:** The actions of the Page Up and Page Down keys are reversed.

***HLPFULL:** Help text is shown on a full display rather than in a window.

***NOSTSMSG:** Status messages are not shown when they are sent to the user.

***STSMSG:** Status messages are shown when they are sent to the user.

***PRTMSG:** A message is sent to this user's message queue when a spooled file for this user is printed or held by the printer writer.

UID Specifies the user ID number (uid number) for this user profile. The uid number is used to identify the user when the user is using the directory file system.

***GEN:** The uid number will be generated for the user. The system will generate a uid number that is not already assigned to another user. The uid number generated will be greater than 100.

user-ID-number: Specify the uid number to be assigned to the user profile. The value can be from 1 to 4294967294. The uid number specified must not already be assigned to another user.

GID Specifies the group ID number (gid number) for this user profile. The gid number is used to identify the group profile when a member of the group is using the directory file system.

***NONE:** The user does not have a gid number.

***GEN:** The gid number is generated for the user. The system generates a gid number that is not already assigned to another user. The gid number generated is greater than 100.

group-ID-number: Specify the gid number to be assigned to the user profile. The value can be from 1 to 4294967294. The gid number specified must not already be assigned to another user.

HOMEDIR

Specifies the path name of the home directory for this user profile. The home directory is the user's initial working directory. The working directory, associated with a process, is used during path name resolution in the directory file system for path names that do not begin with a slash (/). If the home directory specified does not exist when the user signs on, the user's initial working directory is the root (/) directory.

Note: This parameter is only used with the integrated file system. It cannot be used to set the home directory for IBM OS/2 Warp Server for AS/400.

***USRPRF:** The home directory assigned to the user is /home/USRPRF, where USRPRF is the name of the user profile.

'home-directory-path name': Specify the path name of the home directory to assign to this user. For more information on specifying path names, refer to path names.

LOCALE

Specifies the path name of the locale that is assigned to the LANG environment variable for this user.

***SYSVAL:** The system value, QLOCALE, is used to determine the locale path name assigned to this user.

***NONE:** No locale path name is assigned to this user.

***C:** The C locale path name is assigned to this user.

***POSIX:** The POSIX locale path name is assigned to this user.

'locale path name': Specify the path name of the locale assigned to this user. See path names for more information on specifying path names.

AUT Specifies the authority given to users who do not have specific authority to the object, who are not on an authorization list, and whose user group has no specific authority to the object.

***EXCLUDE:** The user cannot access the object.

***ALL:** The user performs all operations on the object except those limited to the owner.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***USE:** You can perform basic operations on the object, such as running a program or reading a file. You cannot change the object. *USE authority provides object operational authority, read authority, and execute authority.

TEXT Specifies the text that briefly describes the user profile named in the USRPRF parameter. More information on this parameter is in Commonly used parameters.

***BLANK:** Text is not specified.

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

Examples for CRTUSRPRF

Example 1: Creating a User Profile

```
CRTUSRPRF   USRPRF(JJADAMS)  PASSWORD(S1CR2T)
  SPCAUT(*SAVSYS)  INLPGM(ARLIB/DSPMENU)
```

This command creates a user profile with the user name of JJADAMS and a password of S1CR2T. After sign-on, a program called DSPMENU in the ARLIB library is called. The user is granted the save system special authority. Because the other parameters were not specified: (1) The profile has no limit on the amount of storage assigned to it for owned permanent objects; (2) A scheduling priority of 3 is the highest priority that any of the user's jobs can have; (3) The user-defined description text is blank; (4) There is no group profile associated with this user profile; and (5) No authority is granted for the user profile to other users.

Example 2: Creating a User Profile with the Same User Name and Password

```
CRTUSRPRF   USRPRF(TMSMITH)  MAXSTG(12)
  INLPGM(PROGMR/CALC)
  TEXT('Ted M. Smith, Dept 410,
  Application Programs')
```

This command creates a user profile with the user name of TMSMITH; the password is also TMSMITH because the password was not specified. The maximum permanent storage space the user can use for all objects is 12K (or 12,288 bytes). The initial program called following sign-on is CALC, which is located in the library named PROGMR. The text parameter provides the user's name, department, and department name. Default values are assigned to the other parameters.

Error messages for CRTUSRPRF

*ESCAPE Messages

CPF22CE
The &1 value &2 is used by another user profile.

CPF22CF
User profile not allowed to be a group profile.

CPF22DB
The user profile being changed must have a GID.

CPF22DF
Unable to process request for user profile &1.

CPF22EB
Unable to process request for user profile &1.

CPF22E1
USROPT parameter cannot specify *STSMSG and *NOSTSMSG.

CPF22F1
Coded character set identifier &1 not valid.

CPF22F3
&1 specified a LMTCPB value that is not permitted.

CPF2202
Do not have authority to create user profile.

CPF2209
Library &1 not found.

CPF2213
Not able to allocate user profile &1.

CPF2214
User profile &1 already exists.

CPF2225
Not able to allocate internal system object.

CPF224A
User profile &1 cannot have a GID and be a member of a group.

CPF2242
Object &1 type *&2 not found in library list.

CPF2244
Object &1 type *&2 cannot be found.

CPF225A
User profile name specified on both USRPRF and SUPGRPPRF parameters.

CPF2259
Group profile &1 not found.

CPF2260
User profile &2 was not created or changed. Reason code &3.

CPF2261
OWNER or GRPAUT value not permitted.

CPF2262
Value for GRPAUT not correct.

CPF2269
Special authority *ALLOBJ required when granting *SECADM or *AUDIT.

CPF2272
Cannot allocate user profile &1.

CPF2291
User profile does not have all special authorities being granted.

CPF2292
*SECADM required to create or change user profiles.

CPF2293
Storage limit exceeded for user profile &1.

CPF9802
Not authorized to object &2 in &3.

CPF9820
Not authorized to use library &1.

CPF9825
Not authorized to device &1.

CRTUDFS (Create User-Defined File System) Command Description

CRTUDFS Command syntax diagram

Purpose

The Create User-Defined File System (CRTUDFS) command creates a file system that can be made visible to the rest of the integrated file system name space via the ADDMFS (Add Mounted File System) or MOUNT command.

A UDFS is represented by the object type *BLKSF, or block special file.

Restrictions:

1. You must have *IOSYSCFG special authority to use this command.
2. The *AUDIT special authority is required when specifying a value other than *SYSVAL on the CRTOBJAUD parameter.

Required Parameter

UDFS Specifies the path name of the file system to be created. It must be in one of the following two forms:

- */dev/QASPXX/udfsname.udfs*, where *XX* is one of the valid system or basic user auxiliary storage pool (ASP) numbers on the system, and *udfsname* is the name of the user-defined file system. All other parts of the name must appear as in the example above.
- */dev/aspname/udfsname.udfs*, where *aspname* is one of the valid independent ASP names on the system, and *udfsname* is the name of the user-defined file system. All other parts of the name must appear as in the example above.

The name part of the path must be unique within the specified *QASPXX* or *aspname* directory.

Optional Parameters

CASE Specifies the case sensitivity of this file system.

***MONO:** The file system will not be case sensitive. For example, the names FileA and filea refer to the same object.

***MIXED:** The file system will be case sensitive. For example, the names FileA and filea do NOT refer to the same object.

DTAAUT

Specifies the public data authority given to the user for the new user-defined file system.

***INDIR:** The authority for the UDFS being created is determined by the directory it is being created in. INDIR for CRTUDFS means the new object will inherit its authorities from the */dev/QASPXX* or */dev/aspname* directory.

***RWX:** The user can change the object and perform basic functions on the object except those limited to the owner or controlled by object existence, object management, object alter and object reference authority. *RWX provides object operational authority and all data authorities.

***RW:** The user can view and modify the contents of an object. *RW authority provides object operational authority and data read, add, update and delete authorities.

***RX:** The user can perform basic operations on the object, such as run a program or display the contents of a file. The user is prevented from changing the object. *RX authority provides object operational authority and read and execute authorities.

***WX:** The user can modify the contents of an object and run a program or search a library or directory. *WX authority provides object operational, and data read, update, delete, and execute authorities.

***R:** The user can view the contents of an object. *R authority provides object operational authority and data read authority.

***W:** The user can modify the contents of an object. *W authority provides object operational authority and data read, update, and delete authorities.

***X:** The user can run a program or search a library or directory. *X authority provides object operational authority and data execute authority.

***EXCLUDE:** The user cannot access the object. The user is prevented from accessing the object. The OBJAUT value must be *NONE, if this special value is used.

***NONE:** The user is given no data authorities to the object. This value cannot be used with OBJAUT value of *NONE.

authorization-list-name: The format of the authorization list name remains the current ten-character format. The OBJAUT value must be *NONE, if this special value is used.

OBJAUT

Specifies the public object authority given to users for the user-defined file system.

***INDIR:** The object authority is based on the authority for the directory where this UDFS is being created. » This means the new UDFS will inherit its primary group, authorization list, and its public, private and primary group authorities from the */dev/qaspXX* or */dev/aspname* directory. « If the value *INDIR is specified for either the OBJAUT parameter or the DTAAUT parameter, then *INDIR must be specified for both parameters.

***NONE:** None of the other object authorities (existence, management, alter or reference) are given to the users. If *EXCLUDE or an authorization list is specified for the DTAAUT parameter, *NONE must be specified. This value cannot be used with the DTAAUT value of *NONE.

***ALL:** All of the other object authorities (existence, management, alter, and reference) are given to the users.

You can specify up to four (4) of the following values:

***OBJEXIST:** The user is given object existence authority to the object. The user can control existence and ownership, delete free storage, perform save/restore operations and transfer ownership of the object.

***OBJMGT:** The user is given object management authority to the object. With this authority you can specify security for the object, move or rename the object and add members to database files.

***OBJALTER:** The user is able to alter the attributes of the objects. On a database file, the user can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. With this authority on an SQL package, the user can change the attributes of the SQL package. Currently, this authority is used only for database files and SQL packages.

***OBJREF:** The user is given object reference authority to objects. Used only for database files, the user can reference an object from another object such that operations on that object may be restricted by the other object. On a physical file, the user can add a referential constraint in which the physical file is the parent.

CRTOBJAUD

Specifies the auditing value of objects created in this UDFS.

***SYSVAL:** The object auditing value for the objects in the UDFS is determined by the system auditing value (QAUDLVL).

***NONE:** Using or changing this object does not cause an audit entry to be sent to the security journal.

***USRPRF:** The user profile of the user accessing this object is used to determine if an audit record is sent for this access. The OBJAUD keyword of the Change User Auditing (CHGUSRAUD) command is used to turn on auditing for a specific user.

***CHANGE:** All change accesses to this object by all users are logged.

***ALL:** All change or read accesses to this object by all users are logged.

DFTFILEFMT

Specifies the format of stream files (*STMF) created in this user-defined file system.

***TYPE2:** A *TYPE2 *STMF has high performance file access and is a new *STMF object format in AS/400 Version 4 Release 4. It will be saved slower than a *TYPE1 *STMF to releases prior to AS/400 Version 4 Release 4. It has a minimum object size of 8192 bytes.

***TYPE1:** A *TYPE1 *STMF has the same format as *STMF objects created on releases prior to AS/400 Version 4 Release 4. It will be saved faster than a *TYPE2 *STMF to releases prior to AS/400 Version 4 Release 4. It has a minimum object size of 4096 bytes.

TEXT Text description for the user-defined file system.

***BLANK:** Text is not specified.

'description': Specify no more than 50 characters, enclosed in apostrophes.

Examples for CRTUDFS

Example 1: Create UDFS in System ASP

```
CRTUDFS  UDFS('/dev/QASP01/joe.udfs) TEXT('Joe Smith')
```

This command creates a user-defined file system (UDFS) named *joe.udfs* in the system auxiliary storage pool (ASP).

Example 2: Create UDFS in ASP 3

```
CRTUDFS  UDFS('/dev/QASP03/harry.udfs') CASE(*MIXED)
```

This command creates a case sensitive user-defined file system (UDFS) named *harry.udfs* in user auxiliary storage pool (ASP) 3.

Error messages for CRTUDFS

None

CRTVLDL (Create Validation List) Command Description

CRTVLDL Command syntax diagram

Purpose

The Create Validation List (CRTVLDL) command creates a validation list. Validation lists contain entries consisting of an identifier, data that will be encrypted by the system when it is stored, and free-form data. Entries can be added, changed, removed, found, and validated. Entries are validated by providing the correct entry identifier and data that is encrypted.

VLDL Specifies the name and library of the validation list being created.

The possible library values are:

***CURLIB:** The validation list is created in the current library. If no library is specified as the current library for the job, QGPL is used.

library-name: Specify the name of the library where the validation list is created.

validation-list-name: Specify the name of the validation list being created.

Optional Parameters

AUT Specifies the authority you are giving to users who do not have specific authority to the object, who are not on an authorization list, and whose group profiles have no specific authority to the object.

The possible values are:

***EXCLUDE:** Exclude authority prevents the user from accessing the object.

***USE:** Use authority allows the user to perform finds and validate entries in the validation list.

***CHANGE:** Change authority allows the user to add, change, remove, find, and validate entries in the validation list.

***ALL:** All authority allows the user to add, change, remove, find, and validation entries in the validation list, and delete the validation list.

authorization-list-name: Specify the name of an authorization list to be used for authority to the object. Users included in the authorization list are granted authority to the object as specified in the list. The authorization list must exist when the object is created.

TEXT Specifies the user-defined text that briefly describes the validation list being created. This text is displayed when the object description is displayed. More information on this parameter is in Commonly used parameters.

***BLANK:** No text is specified.

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

Example for CRTVLDL

```
CRTVLDL  VLDL(WEBLIB/WEBUSRS)  AUT(*EXCLUDE)
  TEXT('My WEB users')
```

This command creates a validation list (WEBUSRS) in the WEBLIB library.

Error messages for CRTVLDL

***ESCAPE Messages**

None.

CRTWSCST (Create Work Station Customizing Object) Command Description

CRTWSCST Command syntax diagram

Purpose

The Create Work Station Customizing Object (CRTWSCST) command allows the user to create a work station customizing object in a library.

Required Parameter

WSCST

Specifies the name of a work station customizing object to be created.

The possible library values are:

***CURLIB:** The work station customizing object is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the work station customizing object is created.

object-name: Specify the name of the object that is created.

Optional Parameters

SRCFILE

Specifies the name of the source file in which a source file member containing the customizing table attributes resides. If the source file does not exist, an error message is displayed. The coded character set identifier for the source file should be *HEX.

***CURLIB:** The source file is located in the current library for the job.

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

QTXTSRC: The IBM-supplied source file QTXTSRC is used.

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

SRCMBR

Specifies the name of the source file member containing the table attributes.

***WSCST:** Specifies that the work station customizing object name is used as the source member name.

source-file-member-name: Specify the name of the member in the source file specified on the SRCFILE parameter that is used to create the work station customizing object.

AUT Specifies the authority given to users who do not have specific authority to the work station customizing object, who are not on an authorization list, and whose user group has no specific authority to the work station customizing object.

***LIBCRTAUT:** The public authority for the work station customizing object is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the work station customizing object). The public authority is determined when the work station customizing object is created. If the CRTAUT value for the library changes after the work station customizing object is created, the new value does not affect any existing objects.

***CHANGE:** The user can perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** The user can perform all operations except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the work station customizing object.

***USE:** The user can perform basic operations on the work station customizing object, such as running a program or reading a file. The user cannot change the work station customizing object. *USE authority provides object operational authority, read authority, and execute authority.

***EXCLUDE:** The user cannot access the work station customizing object.

REPLACE

Indicates whether an existing object is replaced.

***YES:** Replace the existing work station customizing object.

***NO:** Do not replace the existing work station customizing object.

TEXT Specifies the text that briefly describes the object. More information on this parameter is in Commonly used parameters.

***SRCMBRTXT:** The text is taken from the source file member used to create the work station customizing object.

***BLANK:** Text is not specified.

'description': Specify a description of the object.

Example for CRTWSCST

```
CRTWSCST  WSCST(MYLIB/MYWSCOBJ)
```

This command creates a work station customizing object named MYWSCOBJ.

Error messages for CRTWSCST

None

DATA (Data) Command Description

DATA Command syntax diagram

Purpose

The Data (//DATA) command must be used to indicate the beginning of an inline data file in an input stream that is read by a spooling reader. It also specifies the delimiter that must be used to indicate the end of the data file. Inline data files exist only during the current job; they are destroyed after the current job is finished. Unnamed inline files can be used only once in the job.

Restrictions:

1. The DATA command cannot be used from a work station.
2. Two slashes must precede this command name when entering it in the data record, that is, //DATA.
3. The user can separate the slashes from this command name with blank spaces, for example, // DATA.

Optional Parameters

FILE Specifies the name of the inline data file. This name is also specified in the program that processes the file.

QINLINE: The name of the inline data file is QINLINE. The file is processed as an unnamed inline file. An unnamed file can be processed if the program specifies QINLINE as the file name, or if the device file that specifies SPOOL(*YES) is opened for input. Unnamed inline files can be used only once by the job.

inline-file-name: Specify the name of the inline data file used by one or more programs in the job. The file is connected to the program when the program opens the file by specifying its file name. The job can get access to named inline data files more than once.

FILETYPE

Specifies whether the inline data following this command is put in the standard format for source files or in the data file format. The standard source file format is a sequence number (a 6-character source number) followed by the 6-character system date that precedes the data. More information on this parameter is in commonly used parameters.

***DATA:** The diskette file describes data records.

***SRC:** The inline data is numbered in sequence. It is a source file that can be used to create another file or a program.

ENDCHAR

Specifies a string of characters used to indicate the end of an inline data file. To be recognized, the character string must begin in position 1 of the record. If you specify a character string other than by using // (the default value) as the delimiter, all records up to the end-of-file record (the record containing the specified character string starting in column 1) are treated as data. This allows you to imbed reader commands (//BCHJOB, //DATA, //ENDBCHJOB, or //ENDINP) in the data stream. The end-of-file record for ENDCHAR values (which are not default values) is not put to the data file, and it is not checked to see if it is a valid reader command. It is used only to determine the end of the data stream and then it is discarded.

'//': The default value is two slashes. The command works the same way whether two slashes are coded into the parameter or the parameter itself is defaulted.

Using the default, the slashes in positions 1 and 2 of a record (in either a data file or a source file) identify the first record beyond the file. Thus, the commands //BCHJOB, //DATA, //ENDBCHJOB, or //ENDINP also indicate the end of the inline file.

'end-character-string': A character string (up to 25 characters long and enclosed in apostrophes) can be entered to identify the last record in the file. The character string can contain both alphanumeric and special characters. If a character combination other than '//' is specified on the ENDCHAR parameter, reader commands can be safely imbedded in the data. The reader ignores all other data, including reader commands, while searching for the specified string.

IGCDTA

Specifies whether the file contains double-byte character set (DBCS) data.

***NO:** The file does not process DBCS data.

***YES:** The inline file can contain DBCS data.

Examples for DATA

Example 1: Inline Data File in Data File Format

```
//DATA  FILE(FILE1)
```

This command assigns the name FILE1 to the data that follows it, until an end of inline data condition is found (two slashes in positions 1 and 2).

Example 2: Specifying an End Character String

```
//DATA  FILE(FILE2)  ENDCHAR('STOPIT')
```

This command assigns the name FILE2 to the data following it; the file continues until a record is found that contains the characters STOPIT in positions 1 through 6. This delimiter allows the //BCHJOB, //ENDBCHJOB, and //DATA commands and records with // in positions 1 and 2 to be embedded in an inline file.

Example 3: Specifying a File Containing DBCS Data

```
//DATA  FILE(FILE3) IGCDTA(*YES)
```

This command assigns the name FILE3 to the data that follows it. This file can contain DBCS data.

Error messages for DATA

*ESCAPE Messages

CPF1753

Command cannot be run.

DLCOBJ (Deallocate Object) Command Description

DLCOBJ Command syntax diagram

Purpose

The Deallocate Object (DLCOBJ) command releases the allocations of the specified objects. The objects, allocated earlier by one or more Allocate Object (ALCOBJ) commands, are freed for use by other jobs or threads. If the DLCOBJ command is used when a lock does not exist, no error occurs.

If the DLCOBJ command is not used, the objects may be automatically deallocated. Allocated job-scoped locks are automatically released when the job ends. Allocated thread-scoped locks are automatically released when the thread ends. If a thread received a job-scoped lock, the job will continue to hold that lock after the requesting thread ends. » Lock-space-scoped locks are not automatically released. «

The DLCOBJ command should not be issued for an object that was not explicitly allocated by the ALCOBJ command. If the DLCOBJ command is used this way, internal locks on the object are released, making the object capable of being deleted.

Notes:

1. When deallocating distributed data management (DDM) files or distributed files, additional time is required for the command to complete because of the time required for establishing communication and for deallocating files on remote systems.
2. The system does not lock programs when calling them.

Restrictions:

1. This command cannot be used to deallocate a device description, *DEVD, for an Advanced Program-to-Program Communications (APPC) device or for an intrasystem (INTRA) device.
2. This command can be used to deallocate only the following database *FILE types:
   - Physical files
   - Logical files
   - Distributed files
     This deallocates the piece of the file on each node in the node group.
   - DDM files
     This deallocates both the DDM file on the local system and the file on the remote system that is identified in the DDM file.
3. In multithreaded jobs, this command is not threadsafe for distributed files. This command is also not threadsafe and fails for Distributed Data Files (DDM) files of type *SNA.

Required Parameter

OBJ Specifies the qualified name of one or more system objects that are to be deallocated from the job, thread, » or lock space, « the type of each object specified, the lock state of each object, and the member name (if the object is a database file or DDM file).

Element 1: Name of the Object to be Deallocated

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

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

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

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

object-name: Specify the name of the object to be deallocated.

Element 2: Type of Object to be Deallocated

object-type: Specify the type of the object to be deallocated. Refer to Table 1 (101), for the list of object types that are valid for this command.

Element 3: Lock State for the Object to be Deallocated

lock-state: Specify the lock state for the object. Valid lock states include the following:

Value	Lock State Meaning
*SHRRD	Shared for read
*SHRUPD	Shared for update
*SHRNUP	Shared, no update
*EXCLRD	Exclusive, allow read
*EXCL	Exclusive, no read

You can specify all five lock states (*EXCL, *EXCLRD, *SHRUPD, *SHRNUP, and *SHRRD) for most, but not all, object types. Refer to Table 1 (101), for the list of object types and the valid lock states allowed for each object type.

Note: Additional details about lock states can be found in the CL Programming book.

Multiple locks can be specified for the same object in the same job with duplicate or different lock states. Each lock is held separately. For example, if an *EXCL lock is already held for an object, and a second *EXCL lock request occurs, the second lock is also acquired. Both locks must be separately released in the job (deallocated with the DLCOBJ command) before another job can get access to the same object.

Element 4: Member of the Database File to be Deallocated

Note: The following values can only be specified if the object type is a database file.

***FIRST:** The first member of the database file is deallocated.

member-name: Specify the name of member to be deallocated. If the specified file is a logical file, the physical file members associated with the members of the logical file are also deallocated.

Optional Parameter

SCOPE

Specify the scope of the lock request.

***JOB:** The lock is scoped to the job. »

***LCKSPC:**The lock is scoped to the lock space attached to the current thread. If no lock space is attached, the lock is scoped to the job. «

***THREAD:** The lock is scoped to the thread.

All object types supported on the OBJ parameter support job-scoped locks. » All object types supported by the OBJ parameter support lock-space-scoped locks. When deallocating DDM objects with a lock-space-scope, the lock on the remote system scoped to the job is released. « To determine if an object type supports thread-scoped locks refer to Table 1.

The lock scope must match the scope of the locks currently allocated for the job or thread.

Table 1. Object Information

Object Type	Object Type Definition	Lock States[1] *EXCL	*EXCLRD	*SHRUPD	*SHRNUP	*SHRRD	Thread Scope[2]
*AUTL	Authorization List	x	x	x	x	x	
*BNDDIR	Binding directory	x	x			x	
*CLD	C Locale description	x	x	x	x	x	
*CRQD	Change request description	x	x	x	x	x	
*CSI	Communications side information	x	x	x	x	x	
*DEVD	Device Description		x	x			x
*DTAARA	Data area	x	x	x	x	x	x
*DTADCT	Data dictionary	x	x	x	x	x	x
*DTAQ	Data queue	x	x	x	x	x	x
*FCT	Forms control table	x	x	x	x	x	
*FILE	File	x	x	x	x	x	x
*FNTRSC	Font resource	x	x	x	x	x	
*FNTTBL	Font mapping table	x	x	x	x	x	
*FORMDF	Form definition	x	x	x	x	x	
*IMGCLG	Image catalog	x	x	x	x	x	x
*IPXD	Internet packet exchange description	x	x	x	x	x	x
*LIB	Library		x	x	x	x	x
*LOCALE	Locale space object	x	x	x	x	x	x
*MEDDFN	Media definition	x	x	x	x	x	
*MENU	Menu	x	x	x	x	x	
*MGTCOL	Management collection	x	x	x	x	x	x
*MODULE	Module	x	x			x	
*MSGQ	Message queue	x				x	x

*NODL	Node list	x	x	x	x	x	
*NTBD	NetBIOS description	x	x	x	x	x	x
*NWSD	Network server description	x	x	x	x	x	x
*OVL	Overlay	x	x	x	x	x	
*PAGDFN	Page definition	x	x	x	x	x	
*PAGSEG	Page segment	x	x	x	x	x	
*PDG	Print descriptor group	x	x	x	x	x	
*PGM	Program	x	x			x	x
*PNLGRP	Panel group	x	x	x	x	x	
*PSFCFG	Print service facility configuration object	x	x	x	x	x	
*QMFORM	Query management form	x	x	x	x	x	
*QMQRY	Query management query	x	x	x	x	x	
*QRYDFN	Query definition	x	x	x	x	x	
*S36	S/36 machine description	x	x	x	x	x	
*SBSD	Subsystem description	x					x
*SCHIDX	Search index	x	x	x	x	x	
*SQLPKG	Structured Query Language package	x	x	x	x	x	
*SRVPGM	Service program	x	x	x	x	x	x
*SSND	Session description	x	x	x	x	x	
*USRIDX	User index	x	x	x	x	x	x
*USRQ	User queue	x	x	x	x	x	x
*USRSPC	User space	x	x	x	x	x	x
*VLDL	Validation list object	x	x	x	x	x	x
*WSCST	Workstation customizing object	x	x	x	x	x	

[1] x indicates this lock state is allowed for this object type.

[2] x indicates a thread-scoped lock is allowed for this object type.

Examples for DLCOBJ

Example 1: Deallocate a File for the Job

```
DLCOBJ  OBJ((LIBB/FILEA *FILE *SHRRD))
```

This command releases the shared-for-read allocation of the first member of file FILEA in library LIBB that was held by the job.

Example 2: Deallocate a Data Area for the Thread

```
DLCOBJ  OBJ((LIBY/DATAAREAX *DTAARA *SHRRD ))
  SCOPE(*THREAD)
```

This command releases the shared-for-read allocation of the data area DATAAREAX in library LIBY that was held by the thread. »

Example 3: Deallocate File for Lock Space

```
DLCOBJ   OBJ((LIBB/FILEA  *FILE *EXCL MEMBERA)) SCOPE(*LCKSPC)
```

This command deallocates MEMBERA of FILEA in LIBB from the lock space attached to the current thread. «

Error messages for DLCOBJ

***ESCAPE Messages**

CPF1005
Objects not deallocated.

DCL (Declare CL Variable) Command Description

DCL Command syntax diagram

Purpose

The Declare CL Variable (DCL) command defines control language (CL) program variables used in CL programs. CL variables are used to store and update data and to receive parameters from another program on a call. CL variables are known by name only within the program that declares them. They cannot be used outside a CL program except when they are referred to by some commands (such as the DSPPGMVAR command) used for debugging programs. If a variable is declared, but not referred to by another command in a CL program, the variable is not included in the program when it is compiled. However, the value in the variable can be passed to another program as a parameter. Each DCL command defines the attributes of one CL variable and declares its name in the program in which it is used.

Each CL variable in a program must be identified by one of the two declare commands. The Declare File (DCLF) command declares CL variables for display device files and data areas. The DCL command declares all other CL variables.

Restriction: The DCL command is valid only within a CL program. All declare commands (COPYRIGHT, DCL, and DCLF) must follow the PGM (Program) command and must precede all other commands in the program. The three types of declare commands can be intermixed in any order.

Required Parameters

VAR Specifies the name of the CL variable being declared in the CL program. The variable exists only within the program in which it is defined. It can be passed as a parameter on a call to another program, in which case it can be processed by the called program. The variable name must start with an ampersand (&).

TYPE Specifies the type of value contained in the CL variable being declared. The value of the variable

can be a character constant, a decimal constant, or a logical '1' or '0'. The value for this parameter cannot be specified by a CL variable. Specify one of the following types:

***DEC:** A decimal variable that contains a packed-decimal value.

***CHAR:** A character variable that contains a character string value.

***LGL:** A logical variable that contains a logical value of either '1' or '0'.

Optional Parameters

LEN Specifies the length of the CL variable being declared. If the variable is a decimal value, the number of decimal digits to the right of the decimal point can be optionally specified. The value for this LEN parameter cannot be specified by a CL variable. The maximum lengths and the defaults for each of the three types are shown in Table 1 (104).

length: Specify the number of characters that the value in this CL variable can have. The length cannot be greater than the maximum for this type of variable.

decimal-positions: This option is valid only for *decimal* variables. The length of the value in the variable includes the number of decimal positions in the value. The maximum length of the decimal value is 15 digits, including the digits to the right of the decimal point. Up to nine decimal positions can be specified. If nine decimal positions are specified, the value to the *left* of the decimal point can never be greater than 999,999 because only 6 of the 15 digits are left for the integer value.

If a length (in digits) is specified for a decimal variable and the number of decimal positions is not specified, 0 decimal positions is assumed.

VALUE

Specifies the initial value assigned to the CL variable when it is declared in the program. The value must be of the type specified by the TYPE parameter. If no value is specified, a character variable is set to blanks, a decimal variable is set to a value of 0, and a logical variable is set to '0'. The value for the VALUE parameter cannot be specified by a CL variable.

If the name of the declared variable is specified on the PARM parameter of the PGM command in the same program in which the variable is declared, an initial value *cannot* be specified for the variable. In that case, the variable receives its value from the calling program.

Table 1. Variable Lengths

Type	Maximum Length	Default Length[1]
Decimal	15 digits, 9 decimal positions	15 digits, 5 decimal positions
Character	9999 characters[2]	32 characters
Logical	1 character	1 character

[1] For decimal and character types, the default length is the same as the length of the initial value, if one is specified in the VALUE parameter.

[2] The initial value (specified on the VALUE parameter) of a CL variable can be no greater than 3000 characters.

Examples for DCL

Example 1: Specifying the CL Variable Length

```
DCL   &ABLE  *DEC  LEN(5 2)
```

This command declares a CL variable named &ABLE that contains a decimal value. The value can never be greater than 999.99 because LEN specifies up to 5 digits, of which two are to the right of the decimal point. Because the VALUE parameter was not specified, and it is a numeric value, &ABLE is set to a value of zero (000.00).

Example 2: Specifying a Logical Value

```
DCL   &SWITCH   *LGL
```

This command declares a CL variable named &SWITCH to contain a logical value. Because the type parameter specifies logical, the variable is one character long and it is set to '0'.

Example 3: Specifying Initial Value of CL Variable

```
DCL   &FILNAM   *CHAR   VALUE(FILEA)
```

This command declares a CL variable named &FILNAM whose value is FILEA. Because the value contains 5 characters and the LEN parameter was not specified, the length of the variable is also 5 characters.

Error messages for DCL

None

DCLF (Declare File) Command Description

DCLF Command syntax diagram

Purpose

The Declare File (DCLF) command declares one file (by name) to a control language (CL) program. Only one DCLF command is allowed in a CL program. The command specifies the name of the file and the record formats to be used in the program. After the file is declared, the program can contain the data manipulation commands (Send File (SNDF), Receive File (RCVF), Send/Receive File (SNDRCVF), End Receive (ENDRCV), and Wait (WAIT) for display device files, or RCVF for database files) that reference the file. This enables the program to interact with its user by sending data to and receiving data from a work station or database file. Database files can be used for input processing only.

When the CL program is compiled, a CL variable is automatically declared for each field in each record format used in the program. If the file is a nonfield level database file, the record format contains one field with the name of that record format. The field name becomes the variable name prefixed with an ampersand (&). The attributes of each declared field are the same as the attributes of the fields in the file. Fields defined in the record format as numeric are defined as decimal variables. Also, indicators used in the referenced file are declared as logical variables in the form &INnn, in which nn is the indicator number.

Variables automatically declared by the DCLF command can be used in the program the same as the variables declared by a DCL command. For example, indicators can be used in expressions and IF statements because they are declared as logical variables.

The content of the variables, not the variable names, are seen by the user; the display shows one, some, or all of the fields in the record format that can be filled in by the user. DDS determines the display format.

Restrictions:

1. This command is valid only within CL programs. All declare commands (COPYRIGHT, Declare CL Variable (DCL) and Declare File (DCLF)) must follow the Program (PGM) command and precede any other command. The three types of declare commands can be placed in any order.
2. The file must either be a database file with only one record format or be a display file.
3. The file cannot be a mixed file, even if only display devices are defined for that mixed file.
4. The database file can be either physical or logical, and can be either field-level or nonfield level.
5. A file must exist before the program is created.

Because CL variables are automatically declared for each field in a referenced file's record formats, the following restrictions apply:

- If the file is changed (and the file description specifies that level checking is to be performed), the CL program must be recompiled to match the new file description. More information on level checking is in the Database Programming topic in the Information Center and the Application Display Programming

   book.
- If any field name is defined in more than one record format of the display file, the attributes in each record format for the commonly named field must match.
- Any CL variable declared in the program by a DCL command with the same name as an automatically declared CL variable (for a referenced field) must also have the same attributes specified in DDS for the referenced field.
- The variables used in the file must have data types supported for CL variables. However, fields defined as packed decimal format, zoned decimal format, or binary format are declared as decimal variables. The variables used in the file must have attributes that are valid for CL variables. For decimal variables, the limits are 15 digits and 9 decimal positions. For variables with more than 15 digits, CL will declare a character variable that is the number of digits, divided by 2, plus 1 (*n* / 2 + 1) in length. A CPI0306 message will be issued stating that the conversion has taken place, which can be seen in the program listing. For character variables, the limit is 9999 characters.

Required Parameter

FILE Specifies the qualified name of the file used by the CL program. A CL variable name cannot be used to specify the file name.

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

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

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

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

file-name: Specify the name of the file used by the CL program.

Optional Parameters

RCDFMT

Specifies the names of one or more file record formats used by the SNDF, RCVF, and SNDRCVF commands in the CL program. Database files can be processed only by RCVF. CL variable names cannot be specified in RCDFMT; only names of record formats can be used. For every field and indicator in each record format specified in RCDFMT, one CL variable is automatically declared in the program.

***ALL:** Every record format in the file, up to 99, has its fields declared in the CL program as variables. If there are more than 99 record formats in the file, only the first 99 are used.

record-format-name: Specify one or more record format names whose fields are being declared as variables in the CL program. No more than 50 record format names can be specified; CL variables cannot be used to specify the names.

ALWVARLEN

Specifies whether variable length fields are allowed in record formats.

***NO:** Record formats cannot contain variable length fields. A message is sent at compile time if variable-length fields are supported in the file.

***YES:** Record formats can contain variable length fields. CL variables declared for variable-length fields are handled as type *CHAR with length equal to 2 bytes plus the maximum field length. Following a RCVF on a variable-length field, the first 2 bytes in the CL variable contain the length of the data. The data received from the field is padded on the right with blanks to the maximum length allowed (9997 bytes).

ALWNULL

Specifies whether a field value of null is allowed.

***NO:** Values of null are not allowed. For each field containing a null value at RCVF time, a diagnostic message is sent with a single escape message for the entire record. Default values are placed in the CL program variables.

***YES:** Values of null are allowed.

ALWGRAPHIC

Specifies whether graphic data fields are allowed in record formats.

***NO:** Record formats cannot contain graphic data fields. A diagnostic message is sent at compile time if graphic data fields are supported in the file.

***YES:** Record formats can contain graphic data fields. CL variables declared for graphic data fields are handled as type *CHAR with length equal (in bytes) to the graphic data field length.

Examples for DCLF

Example 1: Declaring Fields of All Record Formats as Variables

```
DCLF   FILE(ABLE)
```

This command specifies that the file named ABLE is used by the CL program to pass data between the user and the program. Because no library was specified, the library list is used to locate the file. RCDFMT was not specified; therefore, all the fields and indicators in all the record formats are automatically declared as variables, and data from any field in any record format (up through the first 99) in the file can be passed between the program and the user. No device file routing data is received by the program.

Example 2: Using Multiple Record Formats

```
DCLF   FILE(BAKER)  RCDFMT(REC2 REC6)
```

Display file BAKER is used by the CL program to pass data between the user and the program. Because no library was specified, the library list is used to locate the file. Both the REC2 and REC6 record formats are used. Device file routing data is returned to the program and stored in the CL variable &FB for each RCVF and SNDRCVF command.

Additional Considerations

File processing is handled differently in CL programs, depending on whether the file specified in the DCLF command is a display file or a database file.

The following statements apply if the file is a display device file at compile time:

- The file must be a display device file at run time.
- The program may contain any or all of the following commands: SNDF, RCVF, SNDRCVF, ENDRCV, and WAIT.
- The file is opened for input and output.

The following statements apply if the file is a database file at compile time:

- The file must be a database file with a single record format at run time.
- The program may contain only RCVF commands; SNDF, SNDRCVF, ENDRCV, and WAIT commands are not allowed.
- The file is opened for input only.
- The file is opened by the RCVF command, not the OPNDBF command.

Error messages for DCLF

None

DCPOBJ (Decompress Object) Command Description

DCPOBJ Command syntax diagram

Purpose

The Decompress Object (DCPOBJ) command decompresses programs, panel groups, menus, display files, printer files, modules, and service programs.

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

  Temporary storage is automatically reclaimed when:
  - The RCLTMPSTG command is run
  - The next initial program load (IPL) is run
  - The object is used often enough to cause the system to permanently decompress it

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

Restriction: The user must have *USE authority to the objects specified on the command and execute authority to the library containing the objects.

Required Parameters

OBJ Specifies the qualified name of the object to be decompressed.

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

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

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

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

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

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

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

Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

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

Notes:

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

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

***ALL:** All compressed objects in the specified library are decompressed.

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

object-name: Specify the name of the object that is decompressed.

OBJTYPE

Specifies the type of the object to be decompressed.

***ALL:** All compressed menus, panel groups, display and printer device files, programs, modules, and service programs with the specified object name are decompressed.

***FILE:** The compressed file with the specified object name is decompressed (display and printer files only).

***MENU:** The compressed menu with the specified object name is decompressed.

***MODULE:** The compressed module with the specified object name is decompressed.

***PGM:** The compressed program with the specified object name is decompressed.

***PNLGRP:** The compressed panel group with the specified object name is decompressed.

***SRVPGM:** The compressed service program with the specified object name is decompressed.

Optional Parameter

PGMOPT

Specifies, for *PGM and *SRVPGM objects, the program option that indicates whether the entire program (instruction stream and observability tables) or only the instruction stream is decompressed.

***ALL:** The entire program or service program is decompressed.

***INS:** Only the instruction stream of the program or service program is decompressed.

Example for DCPOBJ

```
DCPOBJ   OBJ(QGPL/*ALL)  OBJTYPE(*FILE)
```

This command decompresses all compressed display and printer files in library QGPL.

Error messages for DCPOBJ

***ESCAPE Messages**

CPF2110
Library &1 not found.

CPF2113
Cannot allocate library &1.

CPF2176
Library &1 damaged.

CPF3B01
Cannot compress or decompress object &1 in &2.

CPF3B02
Cannot compress or decompress file &1 in &2.

CPF3B05
No objects decompressed.

CPF3B06
&1 objects decompressed; &3 not decompressed; &8 not included.

CPF3B08
Cannot allocate object &1 in &2.

CPF8108
Device file or save file &4 in &9 damaged.

CPF812E
Module &4 in &9 damaged.

CPF8129
Program &4 in &9 damaged.

CPF813D
Service program &4 in &9 damaged.

CPF815D
*M36 object &4 in &9 damaged.

CPF815E
*M36CFG object &4 in &9 damaged.

CPF8150
Panel group &4 in &9 damaged.

CPF8151
Menu &4 in &9 damaged.

CPF9570
Error occurred creating or accessing debug data.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9804
Object &2 in library &3 damaged.

CPF9806
Cannot perform function for object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9811
Program &1 in library &2 not found.

CPF9812
File &1 in library &2 not found.

CPF9821
Not authorized to program &1 in library &2.

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

CPF9838
User profile storage limit exceeded.

DLYJOB (Delay Job) Command Description

DLYJOB Command syntax diagram

Purpose

The Delay Job (DLYJOB) command causes the current job to wait for a specified number of seconds, or until a specified time of day, before running resumes.

Required Parameters

DLY Specifies the number of seconds to delay the job. The DLY keyword allows up to 999999 seconds (approximately 11 days, 14 hours). This parameter and the RSMTIME parameter are mutually exclusive.

RSMTIME

Specifies the time of day the job resumes running. When specifying a resume time, the date is implied by the time specified. If the resume time is later than the current time of day, the date is assumed to be the current date. If the resume time is earlier than the current time of day, tomorrow's date is assumed (the delay lasts until that time tomorrow). This parameter and the DLY parameter are mutually exclusive.

Examples for DLYJOB

Example 1: Delaying a Job for 15 Seconds

```
DLYJOB  DLY(15)
```

This command delays running of the job for 15 seconds.

Example 2: Delaying a Job Until Midnight

```
DLYJOB  RSMTIME('000000')
```

This command delays running of the job until midnight tonight.

Error messages for DLYJOB

None

DLTALR (Delete Alert) Command Description

DLTALR Command syntax diagram

Purpose

The Delete Alert (DLTALR) command deletes one or more alerts from the alert data base.

Optional Parameters

DLTOPT

Specifies which alerts are being deleted.

***ALL:** Alerts that meet the selection criteria specified by the remaining parameters are deleted.

***RCV:** Only alerts received from other systems are deleted. Selection criteria for the alerts received can be further specified on the remaining parameters.

***LOCAL:** Only locally created alerts are deleted. Selection criteria for the local alerts can be further specified on the remaining parameters.

***HELD:** Alerts that cannot be sent to the system's focal point and that are marked HELD are deleted. Selection criteria for the held alerts can be further specified on the remaining parameters.

Note: There is a distinction between held alerts that are sent or forwarded by this system, and held alerts that are received by another system. DLTOPT(*HELD) deletes only held alerts that could not be sent or forwarded by this system (alerts that are currently being held based on the ALRHLDCNT network attribute).

DAYS Specifies that alerts older than this value are deleted. Any alert that is more recent than this value is not deleted.

30: Alerts over 30 days old are deleted.

number-of-days: Specify the number of days after which all alerts are deleted.

ALRTYPE

Specifies which types of alerts are deleted. The alert type indicates the severity of the alert.

***ALL:** All types of alerts are deleted.

***TEMP:** Alerts reporting a temporary problem are deleted.

***PERM:** Alerts reporting a permanent problem are deleted.

***PERF:** Alerts reporting a performance problem are deleted.

***IMPEND:** Alerts reporting an impending problem are deleted.

***UNKNOWN:** Alerts reporting a problem with unknown severity are deleted.

alert-type-code-point: Specify the code point for the alert type. The code point is specified by two (2) hexadecimal digits.

ALRRSC

Specifies the name of resources that are reporting problems.

***ALL:** All resources are reporting problems.

resource-name: Specify the names of the resources reporting problems.

ALRRSCTYPE

Specifies the types of resources that are reporting problems. Each resource name has an associated resource type. For example, diskette (DKT) and tape (TAP) are resource types.

***ALL:** All types of resources are reporting problems.

resource-type: Specify the resource type that is reporting problems.

ASNUSER

Specifies the user to which the alerts being deleted are assigned. This value is taken from the value on the ASNUSER parameter in the Add Alert Action Entry (ADDALRACNE) command.

***ALL:** All alerts are deleted.

***NONE:** The alerts not assigned to a user are deleted.

user-name: Specify the name of the user to which the alerts being deleted are assigned.

GROUP

Specifies the group to which the alerts being deleted are assigned. This value is taken from the value on the GROUP parameter in the Add Alert Selection Entry (ADDALRSLTE) command.

***ALL:** All alerts are deleted.

***DEFAULT:** The alerts assigned to the default group are deleted.

***NONE:** The alerts not assigned to a group are deleted.

group-name: Specify the name of the group to which the alerts being deleted are assigned.

Examples for DLTALR

Example 1: Deleting Temporary and Permanent Alert Types

```
DLTALR  DLTOPT(*LOCAL)  DAYS(10)
        ALRTYPE(*TEMP *PERM)  ALRRSCTYPE(DKT)
```

This command deletes temporary and permanent locally created alerts in the alert database. The alerts that are deleted are reporting problems about diskettes. Alerts more than 10 days old that match these criteria are deleted.

Example 2: Deleting Alerts Associated with Diskette Resources

```
DLTALR  DLTOPT(*RCV)  DAYS(0) ALRRSCTYPE(DKT)
```

This command deletes received alerts associated with diskette resources.

Error messages for DLTALR

***ESCAPE Messages**

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9812
File &1 in library &2 not found.

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

CPF9845
Error occurred while opening file &1.

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

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

DLTALRTBL (Delete Alert Table) Command Description

DLTALRTBL Command syntax diagram

Purpose

The Delete Alert Table (DLTALRTBL) command deletes an alert table from the specified library. More information on deleting alert tables is in the Alerts Support  book.

Required Parameters

ALRTBL
Specifies the qualified name of the alert table being deleted.

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

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

alert-table-name: Specify the name of the alert table being deleted.

generic-alert-table-name:* Specify the generic name of the alert table being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Example for DLTALRTBL

```
DLTALRTBL   ALRTBL(ALRTBLLIB/ALRTBLNBR1)
```

This command deletes alert table ALRTBLNBR1 from library ALRTBLLIB.

Error messages for DLTALRTBL

No error messages.

DLTAPARDTA (Delete APAR Data) Command Description

DLTAPARDTA Command syntax diagram

Purpose

The Delete APAR Data (DLTAPARDTA) command deletes the Authorized Problem Analysis Report (APAR) library and any data created by the Restore APAR Data (RSTAPARDTA) command. This command also updates the problem log entry with new information and removes the reference to the APAR library.

Restriction: The following user profiles have authority to this command:

- QPGMR
- QSYSOPR
- QSRVBAS
- QSRV

Required Parameter

PRBID

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

Optional Parameters

ORIGIN

Specifies the network ID and the control point where the problem occurred.

Element 1: Network ID

***NETATR:** The problem originated on a system with the same local network ID as the one defined on the network attributes of this system.

network-ID: Specify the network ID of the system where the problem originated.

Element 2: Control Point Name

***NETATR:** The problem originated on a system with the same control point name as the one defined on the network attributes of this system.

control-point-name: Specify the control point of the system where the problem originated.

Example for DLTAPARDTA

```
DLTAPARDTA  PRBID(9202448748)
```

This command deletes an APAR library and the APAR data for the problem ID 9202448748.

Error messages for DLTAPARDTA

***ESCAPE Messages**

CPF2182

Not authorized to library &1.

CPF39FA

Problem &1 &2 &3 not found

CPF39FE
No APAR data associated with problem &1

CPF39F2
Cannot allocate library &1

CPF39F5
Query of problem &1 failed

CPF39F9
Problem &1 &2 &3 in use

DLTAUTHLR (Delete Authority Holder) Command Description

DLTAUTHLR Command syntax diagram

Purpose

The Delete Authority Holder (DLTAUTHLR) command allows the user to delete an authority holder that secures an object type *FILE. The file must be a program-described database file.

Restriction: You must be the security officer or have all (*ALL) authority to the object to delete the authority holder.

Required Parameter

OBJ Specifies the qualified name of the authority holder to be deleted.

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

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

object-name: Specify the name of the authority holder.

Example for DLTAUTHLR

```
DLTAUTHLR   OBJ(QGPL/FIL1)
```

This command deletes the authority holder for FIL1 in the QGPL library.

Error messages for DLTAUTHLR

***ESCAPE Messages**

CPC2213
Authority holder deleted.

CPF22B1
Authority holder does not exist.

CPF22B2
Not authorized to create or delete authority holder.

CPF9803
Cannot allocate object &2 in library &3.

DLTAUTL (Delete Authorization List) Command Description

DLTAUTL Command syntax diagram

Purpose

The Delete Authorization List (DLTAUTL) command allows a user to delete an authorization list. Authorization lists cannot be deleted if they are being used to secure an object. The user deleting an authorization list must have all (*ALLOBJ) authority or be the owner of the authorization list.

Required Parameter

AUTL Specifies the name or generic name of an authorization list being deleted.

authorization-list-name: Specify the name of the authorization list being deleted.

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

Examples for DLTAUTL

Example 1: Deleting an Authorization List

```
DLTAUTL   AUTL(PROGMR)
```

This commands deletes the PROGMR authorization list.

Example 2: Deleting Generic Authorization Lists

```
DLTAUTL   AUTL(FR*)
```

This commands deletes all authorization lists starting with the letters FR.

Error messages for DLTAUTL

***ESCAPE Messages**

CPF2105
Object &1 in &2 type *&3 not found.

CPF2110
Library &1 not found.

CPF2113
Cannot allocate library &1.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2116
DATA(*YES) specified and *ALL or *FILE not in OBJTYPE list.

CPF2117
&4 objects type *&3 deleted. &5 objects not deleted.

CPF2125
No objects deleted.

CPF2160
Object type *&1 not eligible for requested function.

CPF2176
Library &1 damaged.

CPF2182
Not authorized to library &1.

CPF2189
Not authorized to object &1 in &2 type *&3.

CPF2279
Authorization list &1 cannot be deleted.

CPF2289
Unable to allocate authorization list &1.

CPF5702
File either not DDM file or not found.

CPF9801
Object &2 in library &3 not found.

DLTBNDDIR (Delete Binding Directory) Command Description

DLTBNDDIR Command syntax diagram

Purpose

The Delete Binding Directory (DLTBNDDIR) command deletes a binding directory.

Restriction: You must have *EXECUTE authority for the library where the binding directory is being deleted and *OBJEXIST authority to the binding directory.

Required Parameter

BNDDIR
Specifies a binding directory name or a group of binding directories to be deleted.

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

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

binding-directory-name: Specify the name of the binding directory to be deleted.

generic-binding-directory-name:* Specify the generic name of the binding directory being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, then all binding directories with names that begin with the generic name, and for which the user has authority, are deleted. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete binding directory name.

Example for DLTBNDDIR

```
DLTBNDDIR   BNDDIR(DISPLAYS)
```

This command deletes the binding directory named DISPLAYS from the library list.

Error messages for DLTBNDDIR

*ESCAPE Messages

CPF2105
Object &1 in &2 type *&3 not found.

CPF2110
Library &1 not found.

CPF2113
Cannot allocate library &1.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2117
&4 objects type *&3 deleted. &5 objects not deleted.

CPF2125
No objects deleted.

CPF2160
Object type *&1 not eligible for requested function.

CPF2176
Library &1 damaged.

CPF2182
Not authorized to library &1.

CPF2189
Not authorized to object &1 in &2 type *&3.

CPFA030
Object already in use.

CPFE007
Error occurred processing command.

CPF9803
Cannot allocate object &2 in library &3.

DLTCRQD (Delete Change Request Description) Command Description

DLTCRQD Command syntax diagram

Purpose

The Delete Change Request Description (DLTCRQD) command can be used to delete one or more change request descriptions.

Restriction: You must have object existence authority in order to delete the change request description.

Required Parameter

CRQD Specifies the name and library of the change request description to delete.

change-request-description-name: Specify the name of the change request description to be deleted.

generic-change-request-description-name:* Specify a generic change request description name. A generic name is a character string containing one or more characters followed by an asterisk (*). All change request descriptions with names that match or that begin with the specified string are deleted.

The name of the cluster resource group can be qualified by one of the following library values:

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

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

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

***ALL:** All libraries on the system are searched.

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

Examples for DLTCRQD

Example 1: Deleting a Change Request Description from Your Own Library

```
DLTCRQD CRQD(MYLIB/CHG001)
```

This command deletes a change request description in MYLIB with the name CHG001.

Example 2: Deleting Change Request Descriptions Using a Generic Name

```
DLTCRQD CRQD(*LIBL/CHG*)
```

This command deletes all the change request descriptions in the library list whose names start with CHG.

Example 3: Deleting a Change Request Description from the Current Library

```
DLTCRQD CRQD(*CURLIB/CHG456)
```

This command deletes the change request description named CHG456 in the current library.

Error messages for DLTCRQD

None

DLTCLS (Delete Class) Command Description

DLTCLS Command syntax diagram

Purpose

The Delete Class (DLTCLS) command deletes a class object or a group of class objects from the system. Routing steps that are run by using the class are not affected by its deletion. However, additional routing steps using the deleted class cannot be started. If the deleted class is referred to in an existing routing entry, either the routing entry should be changed (to refer to a different class) or another class with the same name should be created. If the subsystem is active when the class is deleted, errors might occur in the subsystem.

Restriction: The user must have object existence authority for the class being deleted.

Required Parameter

CLS Specifies the name and library of the class descriptions being deleted. A specific class description or a generic class description can be specified; either type can be optionally qualified by a library name.

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

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

QDSNX	» QSYS2xxxxx«	QUSROND
QGPL	QS36F	QUSRPOSGS
QGPL38	QUSER38	QUSRPOSSA
QMPGDATA	QUSRADSM	QUSRPYMSVR
QMQMDATA	QUSRBRM	QUSRRDARS
QMQMPROC	QUSRDIRCL	QUSRSYS
QPFRDATA	QUSRDIRDB	QUSRVI

QRCL	QUSRIJS	QUSRVxRxMx
» QRCLxxxxx «	QUSRINFSKR	
» QSYS2 «	QUSRNOTES	

Notes:

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

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

class-name: Specify the name of the class description being deleted.

generic-class-name:* Specify the generic name of the class. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Example for DLTCLS

```
DLTCLS  CLS(CLASS1)
```

This command deletes the class named CLASS1 from the system.

Error messages for DLTCLS

*ESCAPE Messages

CPF2105
Object &1 in &2 type *&3 not found.

CPF2110
Library &1 not found.

CPF2113
Cannot allocate library &1.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2117
&4 objects type *&3 deleted. &5 objects not deleted.

CPF2160
Object type *&1 not eligible for requested function.

CPF2176
Library &1 damaged.

CPF2182
Not authorized to library &1.

CPF2189
Not authorized to object &1 in &2 type *&3.

CPF5702
File either not DDM file or not found.

DLTCOSD (Delete Class-of-Service Description) Command Description

DLTCOSD Command syntax diagram

Purpose

The Delete Class-of-Service Description (DLTCOSD) command deletes a class-of-service description(s).

Required Parameter

COSD Specifies the name of the class-of-service description to be deleted.

class-of-service-description-name: Specify the name of the class-of-service description to be deleted.

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

Example for DLTCOSD

```
DLTCOSD   COSD(COS01)
```

This command deletes the class-of-service description named COS01 from the system.

Error messages for DLTCOSD

***ESCAPE Messages**

CPF2105
Object &1 in &2 type *&3 not found.

CPF2634
Not authorized to object &1.

DLTCLU (Delete Cluster) Command Description

DLTCLU Command syntax diagram

Purpose

The Delete Cluster (DLTCLU) command deletes a cluster on all nodes currently in the cluster's membership list. All cluster resource group objects and device domains associated with the cluster are also deleted. Cluster Resource Services is ended on each node in the cluster membership list, and the cluster is deleted.

If the command is initiated from a cluster node with a status of Active, all active cluster nodes will be removed from the cluster, and the cluster resource group objects associated with the cluster will be deleted. Cluster resource group objects on nodes with a status of Inactive or Failed will not be deleted. If the command is initiated from a cluster node with a status of Failed or Inactive, only that node is removed from the cluster and cluster resource group objects on that node are deleted.

Cluster resource group exit programs will be called with an action code of Delete (or Delete Command if Cluster Resource Services is not active on the node where the command is invoked).

This command may be called when the cluster is in a partitioned state. In this case, the delete operation will only be performed within the partition running the command.

A node which was a member of a device domain has internal information related to auxiliary storage pools such as disk unit numbers or virtual memory addresses. After a cluster is deleted, this internal information persists until the node is IPLed. If the cluster is deleted, the node must be IPLed before the node can become a member of another device domain.

Restrictions

1. To use this command you must have *IOSYSCFG authority.
2. This CL command must be called from a node defined in the cluster membership list.
3. This command cannot be called from a cluster resource group exit program.

Required Parameter

CLUSTER

Specifies the name of the cluster which is being deleted.

cluster-name: Specify the name of the cluster that is to be deleted.

Example for DLTCLU

```
DLTCLU CLUSTER(MYCLUSTER)
```

This command deletes the cluster MYCLUSTER. All nodes that were in the membership list of MYCLUSTER are no longer members of any cluster.

Error messages for DLTCLU

***ESCAPE Messages**

CPF0001

Error found on &1 command.

DLTCRG (Delete Cluster Resource Group) Command Description

DLTCRG Command syntax diagram

Purpose

The Delete Cluster Resource Group (DLTCRG) command deletes a cluster resource group on the local system only. Deleting a local cluster resource group requires the iSeries Cluster Resource Services to be inactive. The user of this command must have *OBJEXIST authority and *USE authority to the cluster resource group being deleted.

This command will cause the cluster resource group exit program to be invoked with an action code of Delete Command on the local system only. The exit program runs under the user profile specified in the cluster resource group. If the exit program does not exist, a message is logged and the cluster resource group is deleted. The command will never invoke the cluster resource group exit program with an action code of Undo.

Required Parameter

CRG Specifies the name of the cluster resource group that is to be deleted.

cluster resource group name: Name of the cluster resource group that is to be deleted.

Example for DLTCRG

```
DLTCRG CRG(CRGTEST)
```

This command deletes the cluster resource group named CRGTEST from the local system.

Error messages for DLTCRG

*ESCAPE Messages

CPFBB39
Current user does not have IOSYSCFG special authority.

CPFBB41
Cluster resource group &1 exit program ended abnormally.

CPFBB46
Cluster Resource Services internal error.

CPFBB53
Cluster Resource Services is active. Request cannot be processed.

CPF2113
Cannot allocate library &1.

CPF2204
User profile &1 not found.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9804
Object &2 in library &3 damaged.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9872
Program or service program &1 in library &2 ended. Reason code &3.

DLTCRGCLU (Delete Cluster Resource Group from Cluster) Command Description

DLTCRGCLU Command syntax diagram

Purpose

The Delete Cluster Resource Group from Cluster (DLTCRGCLU) command deletes a cluster resource group from all cluster nodes in the recovery domain. The cluster resource group object is marked for deletion and is deleted on each active cluster node. The cluster resource group object will be deleted on other nodes in the cluster when they become active.

Deleting a device cluster resource group will not change the ownership of devices. The devices remain on whatever node owns them at the time of the delete.

If Cluster Resource Services configured the takeover IP address for an application cluster resource group and the IP interface is not active, the takeover IP address will be removed. If Cluster Resource Services finds that the takeover IP address is active, the command will fail.

If an exit program is specified for the cluster resource group, the cluster resource group exit program is called on each active node in the recovery domain with an action code of Verification Phase and action code dependent data of Delete. The cluster resource group status is set to Delete Pending. The cluster resource group will not be deleted if the verification phase call to the exit program fails. This command will not call the cluster resource group exit program with an action code of Undo when the verification phase call to the exit program fails.

The Delete Cluster Resource Group (DLTCRG) command can be used to delete a cluster resource group object on a system that does not have Cluster Resource Services active.

Restrictions

1. To use this command you must have *IOSYSCFG authority.
2. Cluster Resource Services must be active on the node processing the request.
3. The status of the cluster resource group being deleted must not be active.

Required Parameter

CLUSTER
Specifies the name of the cluster containing the cluster resource group.

cluster-name: Specify the name of the cluster.

CRG Specifies the name of the cluster resource group that is to be deleted.

cluster-resource-group-name: Specify the name of the cluster resource group that is to be deleted.

Example for DLTCRGCLU

```
DLTCRGCLU CLUSTER(MYCLUSTER) CRG(MYCRG)
```

This command deletes the cluster resource group named MYCRG from the cluster named MYCLUSTER.

Error messages for DLTCRGCLU

***ESCAPE Messages**

CPF0001
Cluster command &1 failed. See previously listed messages.

DLTCMD (Delete Command) Command Description

DLTCMD Command syntax diagram

Purpose

The Delete Command (DLTCMD) command deletes a user-defined command (or group of commands) from the library where it is located. Only the command definition object is removed; the command definition source, the command processing program, and the validity checker are not affected.

Restriction: The user of this command must have object existence authority for the command being deleted.

Required Parameter

CMD Specifies the qualified name of the commands being deleted. A specific command or a generic command can be specified, and either type can be optionally qualified by a library name.

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

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

1. » 'xxxxx' is the number of a primary auxiliary storage pool.«

2. A different library name, of the form QUSRVxRxMx, can be created by the user for each release that IBM supports. VxRxMx is the version, release, and modification level of the library.

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

command-name: Specify the name of the command being deleted.

generic-command-name:* Specify the generic name of the commands that are being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Example for DLTCMD

```
DLTCMD  CMD(LIB01/PAYROLL)
```

The command named PAYROLL is deleted from library LIB01. Any authorities of use for the command are removed from the profiles of all authorized users.

Error messages for DLTCMD

No error messages.

DLTCSI (Delete Communications Side Information) Command Description

DLTCSI Command syntax diagram

Purpose

The Delete Communications Side Information (DLTCSI) command is used to delete side information objects from the specified libraries.

Required Parameter

CSI Specifies the name of the side information object to be deleted.

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

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

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

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

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

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

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

Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

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

Notes:

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

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

side-information-name: Specify the name of the object that contains the desired side information. This is the CPI-Communications symbolic destination name (sym_dest_name).

generic-side-information-name:* Specify the generic name of the side information object. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Example for DLTCSI

```
DLTCSI   CSI(QGPL/SIDEOBJ)
```

This command deletes the communications side information object named SIDEOBJ from library QGPL.

Error messages for DLTCSI

None

DLTCMNTRC (Delete Communications Trace) Command Description

DLTCMNTRC Command syntax diagram

Purpose

The Delete Communications Trace (DLTCMNTRC) command deletes the communications trace for the specified line, network interface, or network server description. The communications trace can be deleted after the trace has ended.

Restrictions:

1. To use this command you must have *SERVICE special authority, or be authorized to the Service Trace function of Operating System/400 through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
2. The following user profiles have authority to this command:
   - QSECOFR
   - QSRV

Required Parameters

CFGOBJ

Specifies the name of the configuration object that was traced. The object is either a line description or a network interface description.

CFGTYPE

Specifies the type of configuration description that was traced.

***LIN:** The type of configuration object is a line description.

***NWI:** The type of configuration object is a network interface description.

***NWS:** The type of configuration object is a network server description.

Example for DLTCMNTRC

```
DLTCMNTRC  CFGOBJ(*QESLINE)  CFGTYPE(*NWI)
```

This command deletes the communications trace data for line description QESLINE.

Error messages for DLTCMNTRC

***ESCAPE Messages**

CPF2634

Not authorized to object &1.

CPF39A7

Trace storage not available in communications processor

CPF39A8

Not authorized to communications trace service tool

CPF39A9

Error occurred during communications trace function

CPF39B0
No communications traces exist.

CPF39B1
Trace &1 type &2 does not exist

CPF39B2
Trace data for &1 type &2 cannot be deleted

CPF39B6
Communications trace function cannot be performed

DLTCFGL (Delete Configuration List) Command Description

DLTCFGL Command syntax diagram

Purpose

The Delete Configuration List (DLTCFGL) command deletes one or more configuration lists.

Required Parameter

CFGL Specifies the names of the configuration lists being deleted.

configuration-list-name: Specify the name of the configuration list to be deleted. Use the Work with Configuration List (WRKCFGL) command to get a list of configuration lists on the system.

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

Example for DLTCFGL

```
DLTCFGL   CFGL(CONFIG01)
```

This command deletes the configuration list CONFIG01 from the system.

Error messages for DLTCFGL

***ESCAPE Messages**

CPF2105
Object &1 in &2 type *&3 not found.

CPF2625
Not able to allocate object &1.

DLTCNNL (Delete Connection List) Command Description

DLTCNNL Command syntax diagram

Purpose

The Delete Connection List (DLTCNNL) command is used to delete a connection list.

Required Parameter

CNNL Specifies the name of the connection list to be deleted.

connection list-name: Specify the name of the connection list to be deleted.

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

Examples for DLTCNNL

Example 1: Deleting a Specified Connection List

```
DLTCNNL   CNNL(CHICAGO)
```

This command deletes the connection list named CHICAGO. Any entries in connection list CHICAGO are also deleted.

Example 2: Deleting Multiple Connection Lists

```
DLTCNNL   CNNL(CHI*)
```

This command deletes all connection lists whose name begins with CHI. All entries that were listed in the connection lists are also deleted.

Error messages for DLTCNNL

***ESCAPE Messages**

CPF2625
Not able to allocate object &1.

CPF2634
Not authorized to object &1.

CPF266C
Connection list &1 not found.

CPF267D
Connection list &1 not deleted.

CPF268F
Connection list &1 not deleted.

DLTCTLD (Delete Controller Description) Command Description

DLTCTLD Command syntax diagram

Purpose

The Delete Controller Description (DLTCTLD) command deletes controller descriptions.

Required Parameter

CTLD Specifies the name of the controller descriptions being deleted. A specific controller description or a generic controller description is specified.

controller-description-name: Specify the name of the controller description to be deleted.

generic-controller-description-name:* Specify the generic name of the controller description to be deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all

objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. For more information on the use of generic names, refer to generic names.

Example for DLTCTLD

```
DLTCTLD   CTLD(CONTROL01)
```

This command deletes the controller description named CONTROL01 from the system. If the controller description being deleted has any device descriptions associated with it, they are detached and a message containing their names is sent to the system operator.

Error messages for DLTCTLD

***ESCAPE Messages**

CPF2105
Object &1 in &2 type *&3 not found.

CPF2615
Controller &1 must be varied off for this operation.

CPF2634
Not authorized to object &1.

CPF2636
Command not processed. Controller &1 in use.

CPF2697
The request did not complete in the time allotted.

CPF2717
Controller description &1 not deleted.

CPF2782
Message &1 not monitored.

DLTDTAARA (Delete Data Area) Command Description

DLTDTAARA Command syntax diagram

Purpose

The Delete Data Area (DLTDTAARA) command deletes the specified data areas from a library.

Restriction: To use this command, you must have *ALL authority for the data area, and read authority for the library. Local data areas cannot be deleted.

Required Parameter

DTAARA
Specifies the qualified name of the data areas being deleted. A specific data area or a generic data area can be specified; either type can be optionally qualified by a library name.

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

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

data-area-name: Specify the qualified name of the data area being deleted.

generic-data-area-name:* Specify the generic name of the data area being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Example for DLTDTAARA

```
DLTDTAARA    DTAARA(MYLIB/MYDATA)
```

This command deletes the data area named MYDATA from the library MYLIB if the user has the proper authority for the data area and the library.

Error messages for DLTDTAARA

***ESCAPE Messages**

CPF2105
Object &1 in &2 type *&3 not found.

CPF2110
Library &1 not found.

CPF2113
Cannot allocate library &1.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2117
&4 objects type *&3 deleted. &5 objects not deleted.

CPF2176
Library &1 damaged.

CPF2182
Not authorized to library &1.

CPF2189
Not authorized to object &1 in &2 type *&3.

DLTDTADCT (Delete Data Dictionary) Command Description

DLTDTADCT Command syntax diagram

Purpose

The Delete Data Dictionary (DLTDTADCT) command allows a user to delete a data dictionary.

All program-described files linked to definitions in the dictionary must be unlinked before the dictionary can be deleted.

Required Parameter

DTADCT
Specifies the name of the data dictionary being deleted.

Example for DLTDTADCT

```
DLTDTADCT    DTADCT(DEPT547)
```

This command deletes the DEPT547 data dictionary in library DEPT547.

Error messages for DLTDTADCT

***ESCAPE Messages**

CPF2FE2
Dictionary &1 currently in use.

CPF2FE3
System cross reference file is in error.

CPF2FE4
System cross reference file not available.

CPF2F0B
Data dictionary &1 cannot be deleted.

CPF2F0C
Deleting a SQL data dictionary &1 not allowed.

CPF2F0D
Data dictionary &1 partially deleted.

CPF2F0F
Data dictionary &1 not deleted.

CPF2F05
Not authorized to delete dictionary &1.

CPF2F08
Dictionary &1 not found.

CPF9820
Not authorized to use library &1.

DLTDTAQ (Delete Data Queue) Command Description

DLTDTAQ Command syntax diagram

Purpose

The Delete Data Queue (DLTDTAQ) command deletes the specified data queues from the system.

Restriction: The user must have object existence authority for the queue.

Required Parameter

DTAQ Specifies the qualified name of the data queues being deleted. A specific data queue or a generic data queue can be specified, and either type can be optionally qualified by a library name.

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

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

data-queue-name: Specify the name of the data queue being deleted.

generic-data-queue-name:* Specify the generic name of the data queue being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Example for DLTDTAQ

```
DLTDTAQ   DTAQ(DEPTADTAQ)
```

This command deletes the data queue named DEPTADTAQ from the system. The launching point for on-line System API Reference and CL Reference information.

Error messages for DLTDTAQ

No error messages.

DLTIGCDCT (Delete DBCS Conversion Dictionary) Command Description

DLTIGCDCT Command syntax diagram

Purpose

The Delete DBCS Conversion Dictionary (DLTIGCDCT) command erases the specified double-byte character set (DBCS) conversion dictionary from the system. The dictionary contains alphanumeric entries and their related words. The system refers to DBCS conversion dictionaries when doing DBCS conversion.

Restrictions: You must have the following authority to use this command:

- Object operational authorities for this command
- Object operational and object existence authorities for the dictionary
- Object operational authorities to the library in which the dictionary is stored

Required Parameter

IGCDCT

Specifies the qualified name of the DBCS conversion dictionary being deleted. If you do not specify a library name, the system deletes the dictionary found when searching the library list.

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

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

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

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

dictionary-name: Specify the DBCS conversion dictionary being deleted.

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

Example for DLTIGCDCT

```
DLTIGCDCT   IGCDCT(DBCSLIB/IGCDCT*)
```

This command causes the system to delete each DBCS conversion dictionary whose name starts with the characters IGCDCT in library DBCSLIB.

Error messages for DLTIGCDCT

***ESCAPE Messages**

CPF2105

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

CPF2110
Library &1 not found.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2182
Not authorized to library &1.

CPF2189
Not authorized to object &1 in &2 type *&3.

DLTIGCTBL (Delete DBCS Font Table) Command Description

DLTIGCTBL Command syntax diagram

Purpose

The Delete DBCS Font Table (DLTIGCTBL) command deletes the specified double-byte character set (DBCS) font table from the system. DBCS font tables contain the images of extension characters used on the system and are used for displaying and printing these characters in various matrix sizes, depending on the matrix used by the device.

Consider deleting a table that displays and prints characters in a dot matrix pattern not used by your system. Deleting the table saves system storage. Before deleting a table, it can be copied to tape or diskette for future use. To copy a table to tape or diskette, see the CPYIGCTBL (Copy DBCS Font Table) command.

Required Parameter

IGCTBL
Specifies the name of the DBCS font table being deleted. Choose one of the following table names:

QIGC2424: The Japanese DBCS font table used for displaying and printing extension characters in a 24-by-24 dot matrix image.

QIGC2424C: The Traditional Chinese DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC2424K: The Korean DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC2424S: The Simplified Chinese DBCS font table used for printing extension characters in a 24-by-24 dot matrix image.

QIGC3232: The Japanese DBCS font table used for displaying and printing extension characters in a 32-by-32 dot matrix image.

QIGC3232S: The Simplified Chinese DBCS font table used for printing extension characters in a 32-by-32 dot matrix image.

QIGCrrccl: The name of the DBCS font table to be deleted must always be in the format QIGCrrccl, where *rr* is the table row matrix size, *cc* is the table column matrix size, and *l* is an optional language identifier.

Example for DLTIGCTBL

```
DLTIGCTBL   IGCTBL(QIGC2424)
```

This command deletes the Japanese DBCS font table QIGC2424.

Additional Considerations

Do not delete a DBCS font table if any device attached to the system and currently varied on uses that table. If you delete the table, the system sends you a message identifying the following:

- Devices using the deleted tables
- Devices attached to the same controllers such as devices using the table being deleted
- Controllers that will be damaged the next time you try to print or display extension characters on those devices.

If such a problem occurs, do the following:

1. Vary off the affected devices (VRYCFG command).
2. Vary off the affected control unit.
3. Vary on the affected control unit.
4. Vary on the affected devices.
5. Continue normal system work.

For example, do not delete QIGC2424 when a Japanese 5555 display is varied on.

Error messages for DLTIGCTBL

***ESCAPE Messages**

CPF8422
Not able to use DBCS font table &1.

CPF8425
Cancel reply received for message &1.

CPF9830
Cannot assign library &1.

DLTDEVD (Delete Device Description) Command Description

DLTDEVD Command syntax diagram

Purpose

The Delete Device Description (DLTDEVD) command deletes the specified device descriptions.

Restriction: The device identified in the device description must be varied off before this command is sent. If the device description is the last device description on a varied on remote controller description, the remote controller description must be varied off.

Required Parameter

DEVD Specifies the name of the device descriptions being deleted. A specific device description or a generic device description can be specified.

device-description-name: Specify the name of the device description being deleted.

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

Example for DLTDEVD

```
DLTDEVD   DEVD(DSPC01)
```

This command deletes the device description of the device named DSPC01 from the system.

Error messages for DLTDEVD

***ESCAPE Messages**

CPF2105
Object &1 in &2 type *&3 not found.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2117
&4 objects type *&3 deleted. &5 objects not deleted.

CPF2189
Not authorized to object &1 in &2 type *&3.

CPF2605
Not able to allocate &1.

CPF2615
Controller &1 must be varied off for this operation.

CPF2616
Device Description &1 not deleted. Device in use.

CPF2634
Not authorized to object &1.

CPF2646
Delete request failed.

CPF2648
Delete request not allowed for device &1.

CPF2668
Object description not deleted.

CPF268E
Device description &1 cannot be deleted or renamed.

CPF2697
The request did not complete in the time allotted.

CPF2782
Message &1 not monitored.

DLTDKTLBL (Delete Diskette Label) Command Description

DLTDKTLBL Command syntax diagram

Purpose

The Delete Diskette Label (DLTDKTLBL) command deletes the label (that is, the data file identifier) of a named data file from a diskette. The data in the file can optionally be overwritten with binary zeros. If the file is active (the file end date is later than the system date), a message is sent to the system operator. The operator can then either continue to delete the file or end the command.

Note: Unpredictable results can occur when processing diskettes that do not have IBM standard labels. To initialize the diskette, use the Initialize Diskette (INZDKT) command with CHECK(*NO) specified.

Restriction: If diskettes have an extended label area (not supported by the iSeries 400) the extended label area is not searched for the label of the file being deleted.

Required Parameters

DEV Specifies the name of the device in which the diskette being checked is located.

LABEL
Specifies the data file identifier of the file being deleted.

Optional Parameters

VOL Specifies one or more volume identifiers used by the file. More information on this parameter is in Commonly used parameters.

Note: If the CRTDATE parameter is also specified, it must match the date on the file being deleted.

***MOUNTED:** The volume currently placed in the device is used.

volume-identifier: Specify up to 6 characters for the volume identifier of the diskette that contains the file label being deleted; any combination of letters and numbers can be used. If the volume identifiers do not match, a message is sent to the system operator. The operator can then either insert the correct diskette and try again, or stop the command.

CHECK
Specifies whether a check for active files (those with an expiration date later than the system date) is done.

***YES:** If the file is active, an operator message is sent. The operator either can continue or can stop the deletion of the file.

***NO:** The file is deleted without the active file check being done.

CRTDATE
Specifies the creation date of the object.

Note: If CRTDATE is specified and the date on the file being deleted does not match, the file is not deleted and a message is sent to the system operator. The operator can then either try the operation again or stop the command.

***NONE:** No test is made of the date the file was created.

creation-date: Specify the creation date of the file being deleted. The date must be typed in the format specified by the system values QDATFMT or QDATSEP.

OPTION
Specifies how the file is deleted from the diskette.

***SCRATCH:** The expiration date of the file is changed to the current system date. The file can still be referenced for input data. However, when the system writes a new file on the diskette, all files that have expired are deleted to free space for the new files.

***RMV:** The data file identifier is removed from cylinder 0. When the file identifier is gone, the file cannot be referenced for input.

***ERASE:** The data file identifier is deleted from cylinder 0, and data in the file is overwritten with binary zeros.

Examples for DLTDKTLBL

Example 1: Deleting File Without an Active File Check

```
DLTDKTLBL  LABEL(FILEA)  DEV(DKT1)  CHECK(*NO)
```

This command scratches (assumed by the system) FILEA on the diskette in device DKT1 without the active file check.

Example 2: Deleting a File Identifier and Overwriting the Data

```
DLTDKTLBL   LABEL(MONDAY)  DEV(DKT2)  OPTION(*ERASE)
```

This command deletes the file identifier MONDAY from the diskette in device DKT2 and overwrites the data with binary zeros if the file MONDAY is not active (CHECK(*YES) is assumed).

Error messages for DLTDKTLBL

***ESCAPE Messages**

CPF6155
File labeled &1 on device &2 not found.

CPF6156
Cancel reply received for message &6.

CPF6158
Delete diskette label ended; previous error occurred.

CPF6716
Device &1 not a diskette device.

CPF6718
Cannot allocate device &1.

CPF9814
Device &1 not found.

CPF9825
Not authorized to device &1.

DLTDST (Delete Distribution) Command Description

DLTDST Command syntax diagram

Purpose

The Delete Distribution (DLTDST) command allows users to delete their own distributions or distributions for another user if the authority to work on behalf of the other user has been given. This command also allows system operators to delete damaged distribution objects and dangling distribution objects (distribution objects that are not controlled by a distribution tracking object).

Outgoing distributions represent the outgoing confirmation request. This deletes the confirmation entries associated for the particular mail from the confirmation log. The deletion affects only the confirmation request. The request to deliver the mail remains in effect. Also, the request can delete error status information from the undelivered status.

Restrictions:

1. To work on behalf of other users, a user must have special permission given with the Grant User Permission (GRTUSRPMN) command.
2. The requester of the command (the user who is signed on) must be enrolled in the system distribution directory.
3. Personal distribution cannot be deleted if the requester is working on behalf of another user.
4. Damaged and dangling distribution objects can be deleted only by the user who signs on and has *ALLOBJ authority.

Required Parameter

DSTID Specifies the distributions to be deleted.

***ERROBJ:** The distributions to be deleted are damaged or dangling distribution objects.

Note: To delete a damaged or dangling distribution, a value must be specified also on the OBJ parameter.

distribution-identifier: Specify the unique identifier of the distribution to be deleted. The distribution identifier is assigned to the distribution by the system that created it.

Up to 50 distribution IDs can be specified. The distribution identifier consists of the sender's address (padded on the right with blanks up to 8 characters), the sender's user ID (padded on the right with blanks up to 8 characters), and a 4-digit zoned sequence number with leading zeros. For example,

```
'NEWYORK SMITH   0204'   or
UADDRESSUSERIDID0099
```

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

The distribution identifier can be determined with the Query Distribution (QRYDST) command.

Note: The distribution specified cannot be damaged or dangling.

Optional Parameters

OPTION

Specifies whether incoming, outgoing, or error distribution information is deleted.

***IN:** Information about incoming distribution is deleted. An incoming distribution is a distribution being sent to a user. If confirmation of delivery is requested by the sender, notice of the deleted status is returned to the sender.

***OUT:** Information about an outgoing distribution is deleted. An outgoing distribution is the status saved by the system for a distribution that was sent to one or more users with a request for delivery confirmation.

***ERR:** Information about an error or undelivered distribution is deleted. A distribution is sent that was undelivered because an error occurred while routing the distribution.

USRID

Specifies the user ID and address of the user making this request.

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

Element 1: User ID

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

Element 2: User Address

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

DSTIDEXN

Specifies the extension of the distribution identifier specified by the DSTID parameter. This extension identifies duplicate distributions. This extension is a 2-digit extension that ranges from 00 through 99. For incoming distributions, this extension ranges from 01 through 99. For outgoing and undelivered status distributions, this extension is always 00. For example, if the distribution ID is *'NEWYORK SMITH 0204'* and two copies of this distribution were sent to a user, the user has two distributions with the same distribution ID. To distinguish the two distributions, an extension is added to each distribution ID and one extension is *NEWYORK SMITH 020401* and the other one is *NEWYORK SMITH 020402*. If there are no duplicates, the extension defaults to 01. These extensions map one-to-one with the distribution ID specified on the DSTID parameter.

***NONE:** There is no duplicate distribution. This is equivalent to an extension of 00 for outgoing distributions and 01 for incoming distributions.

distribution-ID-extension: Specify the extension associated with the distribution. This is used to identify duplicate distribution IDs. Up to 50 distribution IDs can be specified.

OBJ Specifies the damaged or dangling distribution objects to be deleted.

***NONE:** No damaged or dangling distribution objects are deleted.

***ALL:** All damaged and dangling distribution objects are deleted.

object-name: Specify the name of the damaged or dangling distribution object to be deleted.

The object named can be a local or remote object.

- Local object names consist of the sender's address (padded on the right with blanks to a maximum of 8 characters), the sender's user ID (padded on the right with blanks to a maximum of 8 characters), and a 4-digit zoned sequence number with leading zeros. For example:

  ```
  'NEWYORK SMITH   0204'   or
  UADDRESSUSERIDID0099
  ```

  The apostrophes are needed if there are blanks or special characters in the distribution identifier. The distribution identifier is specified this way because blank characters are valid in a user ID or address.
- Remote object names consist of the remote system name and the system date and time. For example:

  ```
  'QOSRDIST091112509152355'
  ```

  where QOSRDIST is the remote system name, 0911125 is the system date (November 25, 1991), and 09152355 is the system time (9:15:23:55).

The distribution identifier for damaged or dangling distributions can be determined when the Save Document Library Object (SAVDLO) command is run and damaged or dangling distribution objects are identified.

A maximum of 50 objects can be named to be deleted.

CMDCHRID

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

Note: This parameter translates the USRID and DSTID parameters to character set and code page of '930 500'.

The SNA Distribution Services book contains the character set and code page table for '930 500'.

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

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

Element 1: Character Set

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

Element 2: Code Page

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

Examples for DLTDST

Example 1: Deleting Incoming Distribution Sent to User

```
DLTDST  DSTID(('RCH38P  BAKER  0019'))  USRID(*CURRENT)
  DSTIDEXN(02)
```

This command deletes an incoming distribution sent to the user.

Example 2: Deleting Status of Outgoing Distribution

```
DLTDST  DSTID(('SYSTEM20  BRUCE  1361'))  USRID(*CURRENT)
  DSTIDEXN(00)  OPTION(*OUT)  CMDCHRID(101 37)
```

This command deletes the status associated with an outgoing distribution. The DSTID parameter value is encoded with a character identifier where the code page is 37 and the character set is 101. The system performs translation on the DSTID parameter value before trying to locate the distribution on the distribution recipient index.

Example 3: Deleting Incoming Distribution to Signed-On User

```
DLTDST  DSTID(('CHICAGO JONES   0013'))
```

This command deletes an incoming distribution sent to a user who is signed on. The distribution extension defaults to '01' for incoming distributions. If duplicate distributions were sent, this command would only delete the first one.

Example 4: Deleting All Damaged Distributions

```
DLTDST  DSTID(*ERROBJ) OBJ(*ALL)
```

This command deletes all damaged and dangling distributions on the system.

Error messages for DLTDST

***ESCAPE Messages**

CPF89BD
The object must be specified with DSTID(*ERROBJ).

CPF900A
&2 distribution deleted, could not delete &1 distribution.

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

CPF900C
Sign on and verify of user failed.

CPF9016
Request to cancel distribution &2-&3-&4 unsuccessful for &1.

CPF905C
Error occurred trying to find a translation table.

CPF906C
Distribution not deleted.

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

DLTDSTL (Delete Distribution List) Command Description

DLTDSTL Command syntax diagram

Purpose

The Delete Distribution List (DLTDSTL) command deletes an existing distribution list from the system distribution directory. Up to 300 lists can be deleted at a time.

Restriction: You must have security administrator (*SECADM) authority to delete a distribution list that you do not own. No special authority is needed to delete a distribution list that you own.

Required Parameter

LSTID Specifies the two-part list identifier of the distribution list being deleted. Up to 300 list IDs (both parts) can be specified.

Element 1: List Identifier

list-ID: Specify the list identifier (ID) of the distribution list.

Element 2: List Qualifier

list-ID-qualifier: Specify the list ID qualifier of the distribution list.

Note:

The distribution list identifier has two parts, the ID and the qualifier, separated by at least one space. If lowercase characters are specified, the system changes them to uppercase.

The naming rules for the two-part list ID are identical to the rules for the user ID and address. A complete description of these rules is in the SNA Distribution Services  book.

Optional Parameter

CMDCHRID

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

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

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

Element 1: Character Set

graphic-character-set: Specify the graphic character set values used to create the command parameter. Valid values range from 1 through 9999.

Element 2: Code Page

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

Example for DLTDSTL

```
DLTDSTL   LSTID((DEPT48K DLIST)  (ISMGRS DSTLIST))
```

This command deletes the two distribution lists, DEPT48K DLIST and ISMGRS DSTLIST, if they exist.

Additional Considerations

If a specified distribution list identifier does not exist in the directory, an error message is returned. If several list identifiers are given, each is processed individually. An error message is returned each time a list identifier is found not to exist or if the current user of this command does not have the authority to delete the list.

Error messages for DLTDSTL

***ESCAPE Messages**

CPF9024

System cannot get correct record to finish operation.

CPF905C

Error occurred trying to find a translation table.

CPF9094
&1 distribution lists deleted. &2 lists not deleted.

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

CPF9838
User profile storage limit exceeded.

CPF9845
Error occurred while opening file &1.

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

DLTDLO (Delete Document Library Object) Command Description

DLTDLO Command syntax diagram

Purpose

The Delete Document Library Object (DLTDLO) command deletes document library objects from the system.

This command is used to recover disk storage space when system storage becomes critical. It is also used to delete documents and folders that are no longer needed, and generates a request to remove the document from the text index if there is a text entry in the index.

Restrictions:

1. The user of this command must have *ALL, *ALLOBJ, or *SAVSYS authority.
2. Pressing the Enter key deletes the objects immediately; there is no prompting before the document or folder is deleted.
3. A document or folder that is in use is not deleted.
4. When more than one document or folder is specified for deletion, but one of the documents or folders is in use or the user is not authorized to it, a message is sent and the other documents specified for deletion are deleted except the one either being used or restricted by lack of authority.
5. While using this command, the user may encounter an error message indicating that internal objects are locked. This means that another user is using a document library function that cannot run at the same time as the DLTDLO command. Therefore, retry this command later.
6. If a folder, and all documents and folders within it, is being deleted, any document or folder that cannot be deleted from the folder remains, causing the original folders to remain as well; all others are deleted. A message is sent indicating the documents and folders that cannot be deleted.
7. Any combination of FLR, CRTDATE, CHKEXP, DOCCLS, CMDCHRID, and OWNER may be used when DLO(*SEARCH) is specified. When more than one of them is specified, only those documents and folders which meet all of the specified criteria are deleted. Though the FLR parameter can be specified under the preceding criteria, it is not limited to this criteria. The FLR parameter and the SYSOBJNAM parameter are mutually exclusive. It can be specified in any combination except when DLO(*SYSOBJNAM) or the SYSOBJNAM parameter is specified.
8. The document descriptors are searched to locate and delete document library objects that match the search values.
9. CRTDATE contains two lists of two elements each. *N must be specified for an element that precedes the values being specified to maintain that element's position within the parameter value sequence when previous elements in the list are deleted.
10. An ASP value of *ANY can be specified only when DLO(*ALL) FLR(*ANY) or DLO(*SEARCH) FLR(*ANY) is specified.

Required Parameter

DLO Specifies the document library objects (documents and/or folders) that are deleted. If DLO(*SEARCH) is not specified, CRTDATE, CHKEXP, DOCCLS, CMDCHRID, and OWNER cannot be specified.

***ALL:** All documents or folders currently filed in the folder specified in the FLR parameter are deleted. Only the security officer or a user with *SECADM special authority can specify DLO(*ALL) with FLR(*ANY) or FLR(*NONE).

***SEARCH:** All documents or folders currently filed in the document library and are described by the FLR, CRTDATE, CHKEXP, DOCCLS, and OWNER values are deleted.

***SYSOBJNAM:** The system object name for the documents or folders specified in the SYSOBJNAM parameter is used.

document-library-object-name: Specify one or more iSeries 400 document or folder names of specific documents or folders being deleted. Up to 300 documents or folders can be specified. More information on this parameter is in folder and document names.

Optional Parameters

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

Note: More information on folder names is in folder and document names.

***NONE:** The document or folder is not located in a folder. This implies that DLO(*SEARCH) and DLO(*ALL) apply only to the set of all documents and folders not in folders.

***ANY:** A system-wide search is made for the documents to be deleted. This implies that DLO(*SEARCH) and DLO(*ALL) apply to the set of all documents and folders, including those not found in folders.

folder-name: Specify the name of the folder that contains the documents and/or folders being deleted.

CRTDATE

Specifies that documents or folders filed during the specified time period are deleted. If this parameter is specified, then DLO(*SEARCH) must also be specified. For a document or folder being selected, it must have been created on or after the starting time and date, and before or on the ending time and date. This parameter contains two lists of paired elements, one for the starting time and date and one for the ending time and date as follows:

```
((start-time  start-date) (end-time  end-date))
```

If CRTDATE is not specified, the following values are assumed:

```
((*AVAIL *CURRENT) (*AVAIL *END))
```

Element 1: Starting Time

Use one of the following values to specify the starting time. Documents and folders must have been created after this time to be deleted. Documents and folders created before this time are not deleted.

***AVAIL:** Documents and folders created at any time can be deleted.

starting-time: Specify deletion of only those documents and folders that were created at or after the specified time on the starting date. When the start-time is used as a search value, the start-date must not be *BEGIN. The time is specified in 24-hour format with or without a time separator as follows:

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

Element 2: Starting Date

Use one of the following values to specify the starting date. Documents and folders must have been created on or after this date in order to be selected for deletion. Documents and folders created before this date cannot be deleted.

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

***BEGIN:** Documents and folders created on any date can be deleted. The starting date and time is ignored.

starting-date: Specify deletion of only those documents and folders created on or after the date specified. The date must be entered in the format specified by the system value QDATFMT and, if separators are used, QDATSEP.

Element 3: Ending Time

Use one of the following values to specify the ending time. Documents and folders must have been created at or before this time to be deleted. Documents and folders created after the specified time cannot be deleted.

***AVAIL:** Documents and folders created at any time can be deleted.

ending-time: Specify deletion of documents and folders that were created at or before the specified time on the ending date. When the end-time is used as search values, the end-date must not be *END. See the description of *starting-time* for details about how time can be specified.

Element 4: Ending Date

Use one of the following values to specify the ending date. Documents and folders must have been created on or before this date to be deleted. Documents and folders created after this date cannot be deleted.

***END:** Documents and folders created on any date can be deleted. The ending time is ignored when *END is specified.

ending-date: Specify selection of documents and folders created on or before this date. The date must be specified in the format specified by the system value QDATFMT or, if separators are used, QDATSEP.

CHKEXP

Specifies the expiration date of the documents being deleted. The expiration date is assigned to specify when a document is no longer needed. A document with an expiration date (as specified

by the user on the Describe Document display) before the date specified, is deleted. If this parameter is specified, then DLO(*SEARCH) must also be specified.

***NO:** The expiration date for the document is ignored.

***CURRENT:** All documents with an expiration date earlier than today's date are deleted.

document-expiration-date: Specify a date when all documents are to be deleted.

DOCCLS

Specifies the class of documents being deleted. Any character string can be used to specify the document class when creating a document; for example, SPECIFICATION, MEMO, FORM, or PAYROLL. If this parameter is specified, then DLO(*SEARCH) must also be specified.

***ANY:** The DOCCLS parameter is not used to select the documents being deleted. Documents meeting the requirements of the other parameters are deleted.

document-class: Specify a character string ranging from 1 through 16 characters in length to select documents for deletion. The iSeries 400 does not define a meaning for document-class; that is, when creating a document, any character string can be entered into the document class. For comparison, the document class specified and the document class of all documents are changed to uppercase where necessary.

CMDCHRID

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

Application Display Programming  book.

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

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

Element 1: Character Set

graphic-character-set: Specify the graphic character set of the class of documents being deleted. Valid values range from 1 through 999.

Element 2: Code Page

code-page: Specify the code page value used to create the command parameters. Valid values range from 1 through 999.

OWNER

Specifies the owner of the documents and folders being deleted. If this parameter is specified, then DLO(*SEARCH) must also be specified.

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

***ALL:** Documents and folders are selected for deletion without regard to the owners. Only a security officer or a user with *SECADM special authority can specify OWNER(*ALL).

user-profile-name: Specify the name of the user who owns the documents and folders being deleted. Only a security officer or a user with *SECADM special authority is authorized to specify a user profile name other than that belonging to the user who issues this command.

SYSOBJNAM

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

ASP Specifies the identifier (ID) of the auxiliary storage pool (ASP) of the document library object to be

deleted. A value other than *ANY can be specified on this parameter only if *ALL or *SEARCH is specified on the DLO parameter and *ANY is specified on the FLR parameter.

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

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

Note: This option is provided to aid in recovery from damage to a user ASP, particularly one that has overflowed. The document library objects and their details are deleted from the damaged ASP (after the necessary data recovery actions are taken for the ASP), thereby removing overflowed objects and potentially damaged objects from the ASP prior to restoring from backup media. More information on document library objects in ASPs is in the Backup, Recovery, and Availability topic in the Information Center.

Examples for DLTDLO

Example 1: Deleting a Folder

```
DLTDLO   DLO(ABC)
```

This command deletes folder ABC if it contains no documents or folders.

Example 2: Deleting All Documents and Folders Within a Folder

```
DLTDLO   DLO(*ALL)  FLR(ABC)
```

This command deletes all the documents and folders that the user is authorized to delete from folder ABC. If folder ABC turns out to be empty, then it is also deleted.

Example 3: Deleting All Documents in a Document Class, Using *SECADM Special Authority

```
DLTDLO   DLO(*SEARCH)  CRTDATE((*AVAIL 080187)
  (*AVAIL 083187))  DOCCLS(LETTERS)  OWNER(*ALL)
```

This command deletes all documents in document class LETTERS that were filed in the system during August 1987. Only the security officer or a user with *SECADM special authority is allowed to specify OWNER(*ALL).

Example 4: Deleting Current User's Documents

```
DLTDLO   DLO(*SEARCH)  CRTDATE((*AVAIL *BEGIN)
  (*AVAIL *END))  DOCCLS(*ANY)  OWNER(*CURRENT)
```

This command deletes all documents belonging to the current user.

Example 5: Deleting User's Documents Filed Today

```
DLTDLO   DLO(*SEARCH)
```

This command deletes all documents belonging to the specified user that were filed on that day.

Example 6: Deleting Documents and Folders With Document Descriptors

```
DLTDLO   DLO(*SEARCH)  CRTDATE((*AVAIL *BEGIN)
  (*AVAIL *END))  DOCCLS(*ANY)  OWNER(*ALL)
```

This command deletes all documents and folders with document descriptors from the system. If the document descriptors are not synchronized with the document and folder objects, then they are the same as DLO(*ALL). The Backup, Recovery, and Availability topic in the Information Center has more information about synchronizing document descriptors.

Error messages for DLTDLO

***ESCAPE Messages**

CPF2204
User profile &1 not found.

CPF8A16
Document library objects not deleted. &1 objects deleted.

CPF9005
System resource required to complete this request not available.

CPF9006
User not enrolled in system distribution directory.

CPF9012
Start of document interchange session not successful for &1.

CPF9029
Not allowed to specify owner profile &1.

CPF903A
Document or folder activity not stopped, requested operation cannot be done.

CPF9031
No authority to specify DLO(*ALL).

CPF9032
Document interchange session not started.

CPF9046
No documents found satisfying search specification in folder &1.

CPF9062
Date must be specified when time not *AVAIL.

CPF9063
Starting and ending CRTDATE specified in reverse order.

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

CPF9845
Error occurred while opening file &1.

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

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

DLTDOCL (Delete Document List) Command Description

DLTDOCL Command syntax diagram

Purpose

The Delete Document List (DLTDOCL) command deletes a document list from the system. All document lists are stored in library QUSRSYS.

Restrictions:

1. To delete a document list, the user must have object existence authority for the document list, *ALLOBJ special authority, or *SECADM special authority.
2. To delete a document list for another user, the user must have *ALLOBJ or *SECADM special authority.

More information on deleting document list names is in the SNA Distribution Services  book.

Required Parameter

DOCL Specifies the document lists that are being deleted from the system.

***ALL:** All document lists for the specified owner are deleted.

document-list-object-name: Specify the system object name of the iSeries 400 document list that is being deleted. Up to 300 document lists can be specified.

Optional Parameter

OWNER

Specifies the owner of the document lists that are being deleted. This parameter is ignored if a list of document list names is specified on the DOCL parameter.

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

***ALL:** All document lists for all owners are deleted.

user-profile-name: Specify the name of the user who owns the document lists that are being deleted. All document lists owned by this user are deleted.

Examples for DLTDOCL

Example 1: Deleting Current User's Document Lists

```
DLTDOCL   DOCL(*ALL)  OWNER(*CURRENT)
```

This command deletes all document lists owned by the current user.

Example 2: Deleting All Users' Document Lists

```
DLTDOCL   DOCL(*ALL)  OWNER(*ALL)
```

This command deletes all document lists owned by all users.

Example 3: Deleting Document Lists Owned By a Specified User

```
DLTDOCL   DOCL(*ALL)  OWNER(ANN)
```

This command deletes all document lists owned by ANN.

Example 4: Deleting a Specific Document List Owned By User

```
DLTDOCL   DOCL(AN8T475237)
```

This command deletes document list AN8T475237, owned by the user running this command. Displaying the user's owned objects will show the system object name for a specific document list.

Error messages for DLTDOCL

***ESCAPE Messages**

CPF9002
Not allowed to delete specified document lists.

CPF9009
System requires file &1 in &2 be journaled.

CPF9012
Start of document interchange session not successful for &1.

CPF9032
Document interchange session not started.

CPF9051
&1 document lists deleted; &2 document lists not deleted.

CPF9810
Library &1 not found.

CPF9845
Error occurred while opening file &1.

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

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

DLTEDTD (Delete Edit Description) Command Description

DLTEDTD Command syntax diagram

Purpose

The Delete Edit Description (DLTEDTD) command deletes a specified user-defined edit description.

Note: Any data description specifications (DDS), or high-level language programs that have already been created, are not affected.

Required Parameter

EDTD Specifies a single-digit code (5, 6, 7, 8, or 9) or the name of the user-defined edit description being deleted. If a single-digit code is entered, the IBM-supplied edit description name corresponding to the digit is assumed. For example, if the digit 5 is entered, the IBM-supplied edit description name QEDIT5 is assumed.

The possible values are:

single-digit-code: Specify the digit code (5, 6, 7, 8, or 9) for the edit description to be deleted.

edit-description-name: Specify the name of the edit description to be deleted.

The possible library values are:

QSYS: Only the QSYS library is searched for the specified edit description.

library-name: Specify the library to be searched for the specified edit description.

Example for DLTEDTD

```
DLTEDTD   EDTD(5)
```

This command deletes the user-defined edit description 5 from the system.

Error messages for DLTEDTD

***ESCAPE Messages**

CPF2105
Object &1 in &2 type *&3 not found.

CPF2110
Library &1 not found.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2182
Not authorized to library &1.

CPF2189
Not authorized to object &1 in &2 type *&3.

CPF2625
Not able to allocate object &1.

DLTF (Delete File) Command Description

DLTF Command syntax diagram

Purpose

The Delete File (DLTF) command deletes the specified database files or device files from the system. Deleting the files also frees the storage space allocated to the file. If a database file (physical or logical) is deleted, all members contained in the file are also deleted. If the file is in use, the file is not deleted.

A delete operation by generic name deletes every file with that generic name that is eligible to be deleted. For each file that cannot be deleted, a diagnostic message is sent. Files can be ineligible to be deleted for the following reasons:

- Logical files must be deleted before the physical files on which they are based can be deleted.
- SQL views must be deleted before the physical files, SQL tables, or SQL views on which they are based can be deleted.

Restrictions:

1. To delete a file, the user must have object existence authority and object operational for the file and read authority for the library that contains the file.
2. If a physical file is being deleted and a logical file is using the data in the physical file, the logical file must be deleted first.
3. If the DLTF command is entered when debugging and UPDPROD(*NO) was specified on the Start Debug (STRDBG) or Change Debug (CHGDBG) command, a physical file that contains data and is in a production library cannot be deleted.
4. In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA. This command is also not threadsafe and fails for Distributed Data Management (DDM) files of type *SNA, when SYSTEM(*RMT) or SYSTEM(*FILETPYE) is specified.

Required Parameter

FILE Specifies the name of the file being deleted. A specific file name or a generic file name can be specified.

Files that are defined or established as parent or dependent files of a referential constraint can be deleted. If the file being deleted is a dependent file, the following items are also removed:

- All constraint relationships for the dependent file
- All foreign key access paths and foreign keys for the dependent file

If the file being deleted is a parent file, the RMVCST parameter is used to specify the constraint relationships to be removed.

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

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

file-name: Specify the full name of the file being deleted.

generic-file-name:* Specify the generic name of the file being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Optional Parameters

SYSTEM

Specifies whether the file is deleted from the local or a remote system.

***LCL:** The file is deleted from the local system.

***RMT:** The file is deleted from a remote system. A DDM file must be specified on the FILE parameter; the DDM file identifies the file being deleted and the remote system.

***FILETYPE:** The file is deleted from the local system if a DDM file is not specified on the FILE parameter. If a DDM file is specified on the FILE parameter, the remote file identifier in the DDM file is deleted from the remote system.

RMVCST

Specifies how much of the constraint relationships are removed in the associated set of dependent files when you are deleting a parent file of a referential constraint.

***RESTRICT:** The parent file is not deleted and the constraint relationship is not removed if a constraint relationship is defined or established between the parent file and a dependent file. Neither the foreign key access path nor the foreign key of the dependent file is removed.

***REMOVE:** The constraint relationship between the parent file and a dependent file is removed. The corresponding foreign key access path (if one exists and is not shared) and foreign key of a dependent file are removed.

***KEEP:** The constraint relationship is no longer established, but the constraint definition is not removed. Neither the foreign key access path nor the foreign key of the dependent file is removed.

Examples for DLTF

Example 1: Deleting a Specific File

```
DLTF   FILE(BILLING/ORDERS)
```

This command deletes the file named ORDERS in library BILLING. Only the BILLING library is searched for the file.

Example 2: Deleting a Parent File of a Referential Constraint

```
DLTF   FILE(BILLING/QUERIES)  RMVCST(*KEEP)
```

This command deletes the file named QUERIES in the library BILLING. Because the QUERIES file is a parent file of a referential constraint, the established referential constraint is removed, but the definition of the constraint is not removed.

Error messages for DLTF

***ESCAPE Messages**

CPF0601
Not allowed to do operation to file &1 in &2.

CPF0605
Device file &1 in &2 saved with storage freed.

CPF0607
File deleted by another job.

CPF0610
File &1 in &2 not available.

CPF0675
Device file &1 in &2 is in use.

CPF2105
Object &1 in &2 type *&3 not found.

CPF2110
Library &1 not found.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2117
&4 objects type *&3 deleted. &5 objects not deleted.

CPF2182
Not authorized to library &1.

CPF2189
Not authorized to object &1 in &2 type *&3.

CPF2190
Not able to do remote delete or rename request.

CPF320B
Operation was not valid for database file &1.

CPF3203
Cannot allocate object for file &1 in &2.

CPF3220
Cannot do operation on file &1 in &2.

CPF323C
QRECOVERY library could not be allocated.

CPF324B
Cannot allocate dictionary for file &1.

CPF3252
Maximum number of machine locks exceeded.

CPF326A
Operation not successful for file &1 in library &2.

CPF327F
Operation not successful for file &1 in library &2.

CPF3273
File or member not created, deleted or changed.

DLTFTR (Delete Filter) Command Description

DLTFTR Command syntax diagram

Purpose

The Delete Filter (DLTFTR) command deletes a filter object from the specified library.

Required Parameter

FILTER

Specifies the qualified name of the filter being deleted.

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

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

1. » 'xxxxx' is the number of a primary auxiliary storage pool.«

2. A different library name, of the form QUSRVxRxMx, can be created by the user for each release that IBM supports. VxRxMx is the version, release, and modification level of the library.

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

filter-name: Specify the name of the filter being deleted.

generic-filter-name:* Specify the generic name of the filter being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Example for DLTFTR

```
DLTFTR   FILTER(MYLIB/MYFILTER)
```

This command deletes filter MYFILTER from library MYLIB.

Error messages for DLTFTR

None

DLTFRWLOG (Delete Firewall Log) Command Description

DLTFRWLOG Command syntax diagram

Purpose

The Delete Firewall Log (DLTFRWLOG) command deletes firewall log files from the /QIBM/UserData/Firewall/Logs directory.

Restrictions: The user must have the following authorities:

***X** For each directory in the path name preceding the log file(s) to be deleted.

***RWX** For the parent directory of the log file(s) to be deleted.

***OBJEXIST**
For the log file(s) to be deleted.

Required Parameter

NWS Specifies the name of the firewall network server for which log files are to be deleted.

***ALL:** Log files with all firewall names are subject to the deletion criteria.

network-server-name: Specify the name of the firewall whose log files are to be deleted.

Optional Parameter

SLTDATE
Specifies the dates of the log files which are to be deleted. Log files which were written within the selected date range will be deleted.

Element 1: Starting Date

***BEGIN:** The oldest log files through the end date are deleted.

start-date: Specify the date of the first log file to be deleted.

Element 2: Ending Date

***END:** The log files from the start date to the most recent log file are deleted.

end-date: Specify the date of the last log file to be deleted.

Example for DLTFRWLOG

```
DLTFRWLOG NWS(FIREWALL)
        SLTDATE(*BEGIN '12/31/1998')
```

All log files for the firewall on network server FIREWALL that exist through December 31, 1998 are deleted from the system.

Error messages for DLTFRWLOG

No error messages.

DLTFNTRSC (Delete Font Resources) Command Description

DLTFNTRSC Command syntax diagram

Purpose

The Delete Font Resources (DLTFNTRSC) command deletes a font resource from the specified libraries. If the font resource is not found, a message is sent to the user stating that the font resource could not be found.

Required Parameter

FNTRSC

Specifies the qualified name of the font resource.

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

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

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

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

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

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

#CGULIB	#DSULIB	#SEULIB
#COBLIB	#RPGLIB	

#DFULIB #SDALIB

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

font-resource-name: Specify the name of the font resource being deleted.

generic-font-resource-name:* Specify the generic name of the font resource being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Examples for DLTFNTRSC

Example 1: Deleting a Font Resource in the Current Library

```
DLTFNTRSC  FNTRSC(*CURLIB/FNTRSC1)
```

This command deletes the font resource FNTRSC1 if it is in the current library.

Example 2: Deleting Font Resources in All Libraries

```
DLTFNTRSC  FNTRSC(*ALLUSR/FNTRSC1)
```

This command deletes all the font resources named FNTRSC1 in all the user's libraries. Libraries starting with a Q, except for QPGL, are not searched.

Example 3: Deleting Font Resources that Begin with FD

```
DLTFNTRSC  FNTRSC(*LIBL/FD*)
```

This command deletes all the font resources that begin with FD in the job's library list.

Error messages for DLTFNTRSC

None

DLTFNTTBL (Delete Font Table) Command Description

DLTFNTTBL Command syntax diagram

Purpose

The Delete Font Table (DLTFNTTBL) command deletes the specified font table. If the font table is found, it is deleted. If the font table is not found, a message is sent to the user stating that the font table could not be found. Refer to the Printer Device Programming  book for more information on font mapping tables.

Restriction: The PSF/400 feature is required to use this command.

Required Parameter

FNTTBL

Specifies the name of the font table to be deleted.

***PHFCS:** The printer resident to host resident font character set table is to be deleted.

***PHCP:** The printer resident to host resident code page mapping table is to be deleted.

***HPFCS:** The host resident to printer resident font character set table is to be deleted.

***HPCP:** The host resident to printer resident code page mapping table is to be deleted.

The name of a font table must be specified when a printer resident to printer resident font substitution table is to be deleted. The name of the printer resident to printer resident font substitution table can be qualified by one of the following library values:

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

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

library-name: Specify the name of the library where the font table is located.

font-table-name: Specify the name of the printer resident to printer resident font substitution table to be deleted.

Error messages for DLTFNTTBL

None

DLTFORMDF (Delete Form Definition) Command Description

DLTFORMDF Command syntax diagram

Purpose

The Delete Form Definition (DLTFORMDF) command deletes a form definition from the specified library. If the form definition is found, it is deleted. If the form definition is not found, a message is sent to the user stating that the form definition could not be found.

Required Parameter

FORMDF

Specifies the qualified name of the form definition being deleted.

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

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

form-definition-name: Specify the name of the form definition being deleted.

generic-form-definition-name:* Specify the generic name of the form definition. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Examples for DLTFORMDF

Example 1: Deleting the First Occurrence of a Specified Form Definition

```
DLTFORMDF  FORMDF(*LIBL/FORMDF1)
```

This command deletes the first occurrence of FORMDF1 if it was found in the library list.

Example 2: Deleting All Occurrences of a Specified Form Definition

```
DLTFORMDF  FORMDF(*ALLUSR/FORMDF1)
```

This command deletes all the form definitions named FORMDF1 in all user libraries. Libraries beginning with a Q, except for the QGPL library, are not searched.

Example 3: Deleting Form Definitions That Begin With FD

```
DLTFORMDF  FORMDF(*USERLIBL/FD*)
```

This command deletes all the form definitions that begin with FD in all the user libraries.

Error messages for DLTFORMDF

None

DLTFCNARA (Delete Functional Area) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

DLTFCNARA Command syntax diagram

Purpose

The Delete Functional Area (DLTFCNARA) command allows the user to delete functional areas from the system. Functional areas are used by Performance Tools for reports and graphics. A functional area is a pre-defined list of job names and user names that are included in a report or graph.

Required Parameter

FCNARA
Specifies the functional area to be deleted. Enclose the name in apostrophes if it contains any blank spaces between characters.

Optional Parameter

LIB Specifies the library where the functional area is located.

QPFRDATA: The functional area is located in the IBM-supplied performance library, QPFRDATA.

library-name: Specify the name of the library where the functional area is located.

Examples for DLTFCNARA

Example 1: Deleting the Functional Area from the Default Library

```
DLTFCNARA  FCNARA(PERSONNEL)
```

This command deletes the functional area named PERSONNEL from library QPFRDATA.

Example 2: Deleting the Functional Area from a Specified Library

```
DLTFCNARA  FCNARA('Performance Tools')
  LIB(RPFT)
```

This command deletes the functional area named 'Performance Tools' from library RPFT.

Error messages for DLTFCNARA

***ESCAPE Messages**

PFR9067
Cannot delete functional area &2.

DLTGPHFMT (Delete Graph Format) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

DLTGPHFMT Command syntax diagram

Purpose

The Delete Graph Format (DLTGPHFMT) command deletes a graph format. This command also deletes a graph format from any packages that contain it.

Required Parameter

GPHFMT
Specifies the graph format to delete.

The possible library values are:

- **QPFRDATA:** The graph format being deleted is located in the IBM-supplied performance library, QPFRDATA.
- ***CURLIB:** The current library for the job is used to locate the graph format being deleted. If no library is specified as the current library for the job, the QGPL library is used.
- *library-name:* Specify the name of the library where the graph format being deleted is located.

format-name: Specify the name of the graph format being deleted.

Examples for DLTGPHFMT

Example 1: Deleting a Graph Format in a Specified Library

```
DLTGPHFMT  GPHFMT(MYLIB/MYFMT)
```

This command deletes the graph format named MYFMT located in the MYLIB library.

Example 2: Deleting a Graph Format in the Default Library

```
DLTGPHFMT  GPHFMT(TESTFMT)
```

This command deletes the graph format named TESTFMT located in the default library QPFRDATA.

Error messages for DLTGPHFMT

***ESCAPE Messages**

PFR9043
&1 does not exist in library &2.

DLTGPHPKG (Delete Graph Package) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

DLTGPHPKG Command syntax diagram

Purpose

The Delete Graph Package (DLTGPHPKG) command deletes a graph package.

Required Parameter

GPHPKG
Specifies the graph package to delete.

The possible library values are:

- **QPFRDATA:** The graph package is located in the IBM-supplied performance library, QPFRDATA.
- ***CURLIB:** The current library for the job is used to locate the graph package being deleted. If no library is specified as the current library for the job, the QGPL library is used.
- *library-name:* Specify the name of the library where the graph package being deleted is located.

package-name: Specify the name of the graph package to delete.

Examples for DLTGPHPKG

Example 1: Deleting a Package in a Specified Library

```
DLTGPHPKG  GPHPKG(MYLIB/MYPKG)
```

This command deletes the graph package MYPKG from library MYLIB.

Example 2: Deleting a Package in the Default Library

```
DLTGPHPKG  GPHPKG(TESTPKG)
```

This command deletes graph package TESTPKG from default library QPFRDATA.

Error messages for DLTGPHPKG

***ESCAPE Messages**

PFR9043
&1 does not exist in library &2.

DLTGSS (Delete Graphics Symbol Set) Command Description

DLTGSS Command syntax diagram

Purpose

The Delete Graphics Symbol Set (DLTGSS) command deletes a graphics symbol set or group of graphic symbol sets from one or more libraries.

Restriction: The user must have object existence authority for the graphics symbol set being deleted.

Required Parameter

GSS Specifies the qualified name of the graphics symbol set or sets being deleted. A specific graphics symbol set or a generic graphics symbol set can be specified; either type can be optionally qualified by a library name. If no library type is given, *LIBL/ is used to find the graphics symbol set.

The name of the graphics symbol set can be qualified by one of the following library values:

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

QDSNX	» QSYS2xxxxx«	QUSROND
QGPL	QS36F	QUSRPOSGS

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

Notes:

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

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

graphics-symbol-set-name: Specify the name of the graphics symbol set being deleted.

generic-graphics-symbol-set-name:* Specify the generic name of the graphics symbol set that is being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Example for DLTGSS

```
DLTGSS   GSS(ADMUVTIP)
```

This command deletes graphics symbol set ADMUVTIP from the system.

Error messages for DLTGSS

None

DLTHSTDTA (Delete Historical Data) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

DLTHSTDTA Command syntax diagram

Purpose

The Delete Historical Data (DLTHSTDTA) command deletes historical data for selected members from the historical database files.

Optional Parameters

MBR Specifies the member whose historical data is to be deleted.

***SELECT:** All historical database members available in the specified library are listed so the user can select which ones to delete. This value is valid only in an interactive environment.

member-name: Specify the member whose historical data is to be deleted from the database files. Up to 50 members can be specified.

LIB Specifies the library where the files are located.

QPFRDATA: The files are located in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library where the files are located.

JOBD Specifies the job description used to submit jobs for batch processing.

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

- ***LIBL:** All libraries in the job's library list are searched until the first match is found.
- ***CURLIB:** The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.
- *library name:* Specify the name of the library where the job description is located.

QPFRJOBD: The IBM-supplied Performance Tools job description is used.

job-description-name: Specify the name of an alternate job description.

Other Single Values

***NONE:** A batch job is not submitted; instead, processing continues interactively while the user waits. The user's work station cannot be used during this time. This is something to consider for especially long jobs.

Example for DLTHSTDTA

```
DLTHSTDTA   MBR(TEST1)
```

This command deletes historical database file member TEST1 from the historical database files found in the library QPFRDATA. QPFRJOBD is used for the job description of the job.

Error messages for DLTHSTDTA

***ESCAPE Messages**

PFR9033
Too many items selected to be deleted.

PFR9038
Cannot delete historical data from member &2.

PFR9053
Historical data does not exist for member &2.

DLTIMGCLG (Delete Image Catalog) Command Description

DLTIMGCLG Command syntax diagram

Purpose

The Delete Image Catalog (DLTIMGCLG) command is used to delete an image catalog object (*IMGCLG) from library QUSRSYS and optionally delete the associated optical image files. The image files are located in the directory that was specified on the Create Image Catalog (CRTIMGCLG) command. If all the image files are successfully deleted, an attempt will be made to delete the directory which contained the image files.

Restriction:

1. You must have *SECADM and *ALLOBJ special authorities to use this command.

Required Parameter

IMGCLG
Specifies the name of the image catalog to be deleted.

image-catalog-name: Specify the name of the image catalog to be deleted.

Optional Parameter

KEEP Specify whether the optical image files associated with the image catalog are to be kept.

***YES:** Specify that the images associated with the image catalog will be not be deleted.

***NO:** Specify that the images associated with the image catalog will be deleted from the directory which contains the image files.

Examples for DLTIMGCLG

Example 1: Deleting an Image Catalog

```
DLTIMGCLG  IMGCLG(MYCLG)
```

This command deletes image catalog **MYCLG** from library QUSRSYS and leaves the associated image files.

Example 2: Deleting an Image Catalog and the Associated Image Files

```
DLTIMGCLG  IMGCLG(MYCLG) KEEP(*NO)
```

This command deletes image catalog **MYCLG** from library QUSRSYS and all the associated image files.

Error messages for DLTIMGCLG

***ESCAPE Messages**

CPFBC18
Image catalog &1 not deleted.

CPFBC40
Not authorized to command &1.

CPFBC41
&1 command failed.

DLTIPXD (Delete IPX Description) Command Description

DLTIPXD Command syntax diagram

Purpose

The Delete IPX Description (DLTIPXD) command deletes IPX descriptions.

Required Parameter

IPXD Specifies the name of the IPX description being deleted.

IPX-description-name: Specify the name of the IPX description being deleted.

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

Example for DLTIPXD

```
DLTIPXD   IPXD(IPXDESC)
```

This command deletes the IPX Description named IPXDESC from the system.

Error messages for DLTIPXD

***ESCAPE Messages**

CPF26C2
Active IPX description &1 cannot be changed or deleted.

DLTJVAPGM (Delete Java Program) Command Description

DLTJVAPGM command syntax diagram

Purpose

The Delete Java Program (DLTJVAPGM) command deletes an iSeries 400 Java program associated with a Java class file, ZIP file, or JAR file.

If no Java program is associated with the class file specified, informational message JVAB526 is sent and command processing continues.

Restrictions: The class file must be in one of the following file systems: QOpenSys,“root”, or a user-defined file system.

Required Parameter

CLSF Specifies the class file name from which to delete the associated iSeries 400 Java program. The class file name may be qualified by one or more directory names.

class-file-name: Specify the name of the class file or a pattern to match the name(s) of the class file(s) to be used. A pattern can be specified in the last part of the name. An asterisk matches any number of characters and a question mark matches a single character. If the name is qualified or contains a pattern it must be enclosed in apostrophes. An example of a qualified class file name is ’/directory1/directory2/myclassname.class’.

Optional Parameter

SUBTREE

The SUBTREE parameter specifies whether or not subdirectories are processed when looking for files that match the CLSF keyword.

***NONE:** Only the files that match the object name pattern are processed. No subtrees are processed. If the directory has subdirectories, neither the subdirectories nor the objects on the subdirectories are processed.

***ALL:** The entire subtree of the path that is specified in CLSF is processed to create Java programs for files that match the name that was specified on the CLSF parameter.

Example for DLTJVAPGM

Example 1: Delete Java Program

```
DLTJVAPGM CLSF('/projectA/team2/myJavaclassfilename.class')
```

This command will delete the Java program associated with the class file myJavaclassfilename.

Error messages for DLTJVAPGM

***ESCAPE Messages**

JVAB526

Unable to delete Java program for “&1”.

JVAB527

&1 Java programs deleted. &2 Java programs not deleted.

JVAB535

Unmonitored exception received.

DLTJOBD (Delete Job Description) Command Description

DLTJOBD Command syntax diagram

Purpose

The Delete Job Description (DLTJOBD) command deletes the specified job descriptions from the system. Jobs that are running using the specified job description are not affected by its deletion.

Restriction: The user must have object existence authority for the job descriptions being deleted and read authority for the libraries that contain the job descriptions.

Required Parameter

JOBD Specifies the qualified name of the job descriptions being deleted. A specific job description or a generic job description can be specified. Either type can be optionally qualified by a library name.

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

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

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

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

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

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

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

Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

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

Notes:

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

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

job-description-name: Specify the name of the job description being deleted.

generic-job-description-name:* Specify the generic name of the job description being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Example for DLTJOBD

```
DLTJOBD  JOBD(MYLIB/MYJOBD)
```

This command deletes the job description named MYJOBD from library MYLIB.

Error messages for DLTJOBD

***ESCAPE Messages**

CPF2105
Object &1 in &2 type *&3 not found.

CPF2110
Library &1 not found.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2117
&4 objects type *&3 deleted. &5 objects not deleted.

CPF2182
Not authorized to library &1.

CPF2189
Not authorized to object &1 in &2 type *&3.

DLTJOBQ (Delete Job Queue) Command Description

DLTJOBQ Command syntax diagram

Purpose

The Delete Job Queue (DLTJOBQ) command deletes the specified job queue or queues from the system.

Restrictions:

1. The job queue being deleted cannot contain any entries; all jobs on the queue must be completed, deleted, or moved to a different job queue.
2. A subsystem cannot be active to the job queue.

Required Parameter

JOBQ Specifies the (optionally) qualified name of the job queue or queues being deleted. A specific job queue name or a generic job queue name can be specified and either type can be optionally qualified by a library name.

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

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

job-queue-name: Specify the name of the job queue being deleted

generic-job-queue-name:* Specify the generic name of the job queue being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

The libraries searched for the job queues being deleted depend on the library qualifier that is specified or assumed.

Example for DLTJOBQ

```
DLTJOBQ  JOBQ(SPECIALJQ)
```

This command deletes the job queue SPECIALJQ from the system.

Error messages for DLTJOBQ

***ESCAPE Messages**

CPF1763
Cannot allocate one or more libraries.

CPF2105
Object &1 in &2 type *&3 not found.

CPF2110
Library &1 not found.

CPF2117
&4 objects type *&3 deleted. &5 objects not deleted.

CPF2182
Not authorized to library &1.

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

CPF3324
Job queue &1 in &2 not deleted. Job queue in use.

CPF3330
Necessary resource not available.

DLTJRN (Delete Journal) Command Description

DLTJRN Command syntax diagram

Purpose

The Delete Journal (DLTJRN) command deletes the specified journal or journals from the system.

Restrictions:

1. Objects cannot be journaled to the specified journal, nor can any job that used this journal for commitment control still be active, when this command is issued. To determine if any objects are being journaled, issue the Work with Journal Attributes (WRKJRNA) command.
   - If any objects are being journaled, issue the End Journal Access Path (ENDJRNAP), End Journal (ENDJRN), End Journal Object (ENDJRNOBJ), and End Journal Physical File Changes (ENDJRNPF) commands to end journaling.
   - If any job that used this journal for commitment control is still active, issue the End Job (ENDJOB) command for each active job.
2. A journal cannot be deleted if it is actively receiving journal entries from its source journal or replicating journal entries to another remote journal. To determine the remote journaling information for a journal, use the Work with Journal Attributes (WRKJRNA) command. For more information regarding remote journaling, see the » Journal management article in the Information Center.«

Required Parameter

JRN Specifies the qualified name of the journal or journals to be deleted. A specific journal name or a generic journal name can be specified; either type of journal name can be optionally qualified by a library name.

Depending on the library name that is specified or assumed, the following libraries are searched for the journals to be deleted:

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

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

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

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

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

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

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

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

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

Notes:

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

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

journal-name: Specify the qualified name of the journal being deleted.

generic-journal-name:* Specify the generic name of the journal being deleted. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be deleted only if *ALL or *ALLUSR library values can be specified for the name. For more information on the use of generic names, refer to generic names.

Example for DLTJRN

```
DLTJRN   JRN(MYLIB/JRNLA)
```

This command deletes the journal named JRNLA in library MYLIB from the system.

Error messages for DLTJRN

*ESCAPE Messages

CPF2105
Object &1 in &2 type *&3 not found.

CPF2110
Library &1 not found.

CPF2113
Cannot allocate library &1.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2117
&4 objects type *&3 deleted. &5 objects not deleted.

CPF2125
No objects deleted.

CPF2153
Journal QAUDJRN in library QSYS not deleted.

CPF2154
Objects of type authority holder cannot be displayed.

CPF2160
Object type *&1 not eligible for requested function.

CPF2176
Library &1 damaged.

CPF2182
Not authorized to library &1.

CPF2189
Not authorized to object &1 in &2 type *&3.

CPF70EA
Cannot delete journal &1 in &2.

CPF70E1
Cannot delete journal &1 in &2.

CPF701B
Journal recovery of an interrupted operation failed.

CPF7021
Cannot delete journal &1 in &2.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9830
Cannot assign library &1.

Printed in U.S.A.