



@server

iSeries

CL Commands Volume 15





@server

iSeries

CL Commands Volume 15

Contents

Command Descriptions	1
MRGTCPHT (Merge TCP/IP Host Table) Command Description	1
MGRBRM (Migrate Using BRM) Command Description	3
MONMSG (Monitor Message) Command Description	3
MONSWABRM (Monitor Save While Active using BRM) Command Description	6
MOV (Move) Command Description	6
MOVE (Move) Command Description	11
MOVDOC (Move Document) Command Description	11
MOVMEDBRM (Move Media Using BRM) Command Description	12
MOVOBJ (Move Object) Command Description	12
MOVSPLFBRM (Move Saved Spooled Files Using BRM) Command Description	19
NETSTAT Command	19
OPNQRYF (Open Query File) Command Description	20
ORDSPTPTF (Order Supported Product PTFs) Command Description	60
OVRDBF (Override with Database File) Command Description	61
OVRDKTF (Override with Diskette File) Command Description	71
OVRDSPF (Override with Display File) Command Description	78
OVRICFF (Override with Intersystem Communications Function File) Command Description	84
OVRICFDEVE (Override with Intersystem Communications Function Program Device Entry) Command Description	88
OVRMSGF (Override with Message File) Command Description	98
OVRPRTF (Override with Printer File) Command Description	99
OVRSAVF (Override with Save File) Command Description	129
OVRTAPF (Override with Tape File) Command Description	132
PKGINSOBJ (Package Installable Object) Command Description	149
PKGPRDDST (Package Product for Distribution) Command Description	156
PKGPRDOPT (Package Product Option) Command Description	160
PING Command	162
POSDBF (Position Database File) Command Description	162
PWRDWNSYS (Power Down System) Command Description	163
PRTACTRPT (Print Activity Report) Command Description	168
PRTADPOBJ (Print Adopting Objects) Command Description	172
PRTAFPDTA (Print Advanced Function Printer Data) Command Description	173
PRTCMDUSG (Print Command Usage) Command Description	176
PRTCINSEC (Print Communications Security) Command Description	178
PRTCINTRC (Print Communications Trace) Command Description	178
PRTCPTRPT (Print Component Report) Command Description	184
PRTDEVADR (Print Device Addresses) Command Description	193
PRTDSKINF (Print Disk Information) Command Description	193
PRTDOC (Print Document) Command Description	196
PRTERRLOG (Print Error Log) Command Description	211
PRTINTDTA (Print Internal Data) Command Description	217
PRTIPSCFG (Print IP over SNA Configuration) Command Description	219
PRTJOBDAUT (Print Job Description Authority) Command Description	220
PRTJOBTRPT (Print Job Report) Command Description	222
PRTJOBTRC (Print Job Trace) Command Description	229
PRTLBLBRM (Print Labels Using BRM) Command Description	232
PRTLCKRPT (Print Lock Report) Command Description	233
PRTMEDBRM (Print Media Exceptions Using BRM) Command Description	236
PRTMOVBRM (Print Media Movement Using BRM) Command Description	236
PRTPEXRPT (Print Performance Explorer Report) Command Description	237
PRTTCPPTP (Print Point-to-Point TCP/IP Profile) Command Description	244
PRTPOLRPT (Print Pool Report) Command Description	245

PRTPVTAUT (Print Private Authorities) Command Description	253
PRTPRFINT (Print Profile Internals) Command Description.	255
PRTPUBAUT (Print Publicly Authorized Objects) Command Description	255
PRTQAUT (Print Queue Authority) Command Description	258
PRTRSCRPT (Print Resource Report) Command Description.	260
PRTSCDJS (Print Schedule using Job Scheduler) Command Description	264
PRTSWL (Print Stop Word List) Command Description	265
PRTSQLINF (Print Structured Query Language Information) Command Description.	265
PRTSBSDAUT (Print Subsystem Description Authority) Command Description	266
PRTSYSINF (Print System Information) Command Description	268
PRTSYSRPT (Print System Report) Command Description.	269
PRTSYSSECA (Print System Security Attributes) Command Description	276
PRTTRC (Print Trace) Command Description.	276
PRTTRCRPT (Print Trace Report) Command Description	279
PRTTNSRPT (Print Transaction Report) Command Description	282
PRTTRGPGM (Print Trigger Program) Command Description	287
PRTUSROBJ (Print User Objects) Command Description	288
PRTUSRPRF (Print User Profile) Command Description.	289
PGM (Program) Command Description	291
QSH (Start QSH) Command	292
QRYDST (Query Distribution) Command Description	293
QRYDOCLIB (Query Document Library) Command Description	297
QRYPRBSTS (Query Problem Status) Command Description.	308
QRYTIEF (Query Technical Information Exchange File) Command Description	310

Command Descriptions

MRGTCPHT (Merge TCP/IP Host Table) Command Description

MRGTCPHT Command syntax diagram

Purpose

The Merge TCP/IP Host Table (MRGTCPHT) command is used to merge host names, internet addresses, and text comment entries from a physical file member into the local host table. A replace option is also provided that allows the entire local host table to be replaced by the host table entries in a user specified physical file member.

A file format option is provided that allows either *AS400, *AIX, or *NIC file format to be merged with the local host table. The local host table is located in member QUSRSYS/QATOCHOST.HOSTS and is created as a physical file.

A maximum of 4 host names per IP address is allowed when host tables are merged. For example: If the local host table already has 3 host names and the physical file member to be merged has 2 additional host names, only the first host name in the physical file is merged into the final host table. Host names that exist both in the local host table and the physical file member being merged are not duplicated.

Attention:

The original copy of the local host table is not saved by the Merge TCP/IP Host Table (MRGTCPHT) command. To save the original host table, create a copy of the file QUSRSYS/QATOCHOST.HOSTS using the Copy File (CPYF) command. Do this before issuing the MRGTCPHT command.

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

Required Parameter

FROMFILE

Specifies the name of the physical file that contains the member being used for the merge operation. If no library qualifier is specified, the library list (*LIBL) is used to locate the file.

The name of the physical file member 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.

from-file: Specify the name of the physical file.

Optional Parameters

FROMMBR

Specifies the name of the physical file member to be used in the merge operation.

***FIRST:** The first member of the physical file is used to merge with the host table.

***LAST:** The last member of the physical file is used to merge with the host table.

from-member: Specify the name of the physical file member.

FILEFMT

Specifies whether the physical file member to be merged with the local host table is *AS400, *AIX, or *NIC format.

***AS400:** The physical file member to be merged with the local host table is *AS400 format.

Note:

*AS400 can only be used if the physical file member specified is a host table from an AS/400 system running Version 3, Release 1, Modification 0 (V3R1M0) or later of OS/400. If you import a host table from an AS/400 system running any version of OS/400 prior to V3R1M0, specify *AIX as the FILEFMT.

***AIX:** The physical file member to be merged with the local host table is *AIX format.

***NIC:** The physical file member to be merged with the local host table is *NIC format.

REPLACE

Specifies whether the physical file member is to be merged with or replaces the local host table.

***NO:** The physical file member is merged with the local host table.

***YES:** The physical file member replaces the local host table.

Examples for MRGTCPTH

Example 1: Replacing Local Host Table

```
MRGTCPTH FROMFILE(AS400FILE) REPLACE(*YES)
```

This command replaces the contents of QUSRSYS/QATOC HOSTS with the contents of the first member of physical file AS400FILE. The first member of physical file AS400FILE is in AS/400 host table format.

Example 2: Merging Local Host Table

```
MRGTCPTH FROMFILE(HOSTLIB/NICFILE)  
FROMMBR(NEWHOSTS) FILEFMT(*NIC)
```

This command merges the current contents of the local host table with the contents of the NEWHOSTS member of physical file NICFILE in library HOSTLIB. The physical file is in *NIC format. The records are converted from *NIC format to *AS400 format by this command.

Error messages for MRGTCPTH

*ESCAPE Messages

TCP1927

Records of file &1, member &2 not valid.

TCP1929

Host table not available.

TCP1934

Merge file &1, member &3, in library &2 not found.



MGRBRM (Migrate Using BRM) Command Description

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

MGRBRM Command syntax diagram

Purpose

The Migrate using BRM (MGRBRM) command allows you to migrate a library or first level folder to a specified ASP. This command is used by BRMS migration processing and by you to request migration of a specific library or folder as needed. You can specify the ASP to which you want to migrate the library or folder.

When the MGRBRM command is used, BRMS ignores low storage threshold constraints for the ASP from which the item is being moved, but does honor the high storage threshold for the target ASP.

Restriction: The Advanced Functions feature is required to use this command.

Example for MGRBRM

Example 1: Migrating a Library

```
MGRBRM  
TOASP(COMPRESS) TYPE(*LIB) LIB(GLLIB)
```

In this example the library GLLIB is to be migrated to the auxiliary storage pool named COMPRESS.

Error messages for MGRBRM

No error messages.

MONMSG (Monitor Message) Command Description

MONMSG Command syntax diagram

Purpose

The Monitor Message (MONMSG) command is used to monitor escape, notify, and status messages sent to the program message queue of the program in which the command is being used. Completion and diagnostic messages cannot be monitored.

When the MONMSG command is compiled in a control language (CL) program, it establishes a monitor for the arrival of the specified messages. The command monitors the messages for the condition specified by the comparison data given in the command. If a message meeting the conditions arrives on the message queue, the CL command specified on the MONMSG command is processed.

Up to 1000 MONMSG commands can be specified in a program to monitor the arrival of messages for specific conditions or for a group of conditions. Specific message identifiers or generic message identifiers can be monitored.

The MONMSG command can be coded following most commands in a CL program. A MONMSG command that is not placed at the beginning of the program applies only to the immediately preceding

command; this is called a command-level MONMSG command. The command-level MONMSG command monitors only messages sent by the previous command. If the message sent by that command meets the conditions specified in the MONMSG command, the action specified in the same MONMSG command is taken. As many as 100 MONMSG commands, coded immediately after a command, can monitor the messages sent by that command.

When the action specified in the MONMSG command has been performed, and that action does not end with a GOTO or RETURN command, control returns to the command in the program that follows the command that sent the message. If the action ends with a GOTO command, control branches to the command in the program specified in the GOTO command. If the action ends with a RETURN command, control returns to the program that called the program that contains the MONMSG command.

If one or more MONMSG commands are placed at the beginning of the program, immediately following the declare commands or the PGM command if there are no declare commands, they monitor messages sent by all of the commands in the program (maximum of 100). This is called a program-level MONMSG command. If any message sent by any command in the program meets the conditions specified in any one of the program-level MONMSG commands, the corresponding action specified in the same command is taken.

The action taken by a command-level MONMSG command overrides a program-level MONMSG command.

If a command is coded for the EXEC parameter on a MONMSG command that is placed at the beginning of a program, *only* the GOTO command can be used, and it must specify the label for the command to which control is to be passed if a monitored message occurs. If a command is not coded for the EXEC parameter, monitored messages are ignored.

Restrictions:

1. This command is valid only in CL programs.
2. It can be coded after the last declare command (if declare commands are used), following the PGM command that begins the program, or it can be coded following any command allowed in CL programs, except for the following: DO, ELSE, ENDDO, ENDPGM, GOTO, IF, or RETURN. Note that if another program sends a message that is monitored by this command, a return cannot be made to that program.

Required Parameter

MSGID

Specifies the message identifiers of one or more escape, notify, or status messages that are monitored by this command. As many as 50 specific and/or generic message identifiers can be specified on one command.

Note:

Many CL commands issue one escape message for many different error conditions. Details about the error or failure are given in diagnostic messages that precede the escape message. Although diagnostic messages cannot be monitored, they can be received from the job's external message queue after the escape message has activated the user's message monitor.

The first 3 characters must be a code consisting of an alphabetic character followed by 2 alphanumeric (alphabetic or decimal) characters; the last 4 characters can consist of the decimal numbers 0 through 9 and the characters A through F.

Note: Message identifiers using the MCH code (MCHnnnn) use only the numbers 0 through 9 in the last four characters.

If zeros are specified in either two or all four of the rightmost positions, such as ppmm00, a generic message identifier is specified. For example, if CPF0000 is specified, all the CPF messages are monitored.

Specify the message identifiers of 1 to 50 messages that are monitored when they arrive at this program's message queue. CL variables cannot be used to specify message identifiers.

Optional Parameters

CMPDTA

Specifies the comparison data used to determine whether the monitored message (having one of the specified message identifiers) received on the program's message queue is acted on by this command. The message data specified in the MSGDTA parameter of the Send Program Message (SNDPGMMSG) command is compared with this comparison data. If the first part (up through the first 28 characters, or less) of the message's substitution values matches the comparison data specified, the action specified in the EXEC parameter of this command is taken. The action is also taken if no comparison data is specified.

***NONE:** No comparison data is specified; if the message in the program's message queue is from a command that this command is monitoring, and if it has the specified identifier, the action specified by EXEC is taken.

comparison-data: Specify a character string of no more than 28 characters, enclosed in apostrophes if necessary, that is compared with the same number of characters in the message data of the received message, starting with the first character in the message data. If the comparison data matches the first part of the received message data, this command performs the function specified in the EXEC parameter. A CL variable cannot be specified for the comparison data.

The comparison data can be displayed by the Display Program Variable (DSPPGMVAR) command.

EXEC Specifies the CL command that is processed when a monitored message sent to the program's message queue meets the conditions specified in this MONMSG command. If no command is specified and a monitored message arrives on the queue, the message is ignored, and control passes to the next command in the program.

If the MONMSG command is placed at the beginning of the program, the EXEC parameter must specify the GOTO command and the label identifying the command that receives control.

Specify the CL command, including its parameters to be used, that is run when a message meeting the conditions specified in this command is received. The command specified is not run if the received message does not meet the specified conditions. A CL variable cannot be specified in place of the CL command.

Note: If a DO command is specified on EXEC, the entire DO group associated with the DO command is run if the condition is met.

Examples for MONMSG

Example 1: Monitoring Messages Sent by any Command

```
PGM
MONMSG MSGID(CPF0001 CPF1999) EXEC(GOTO EXIT2)
```

This example shows a MONMSG command at the beginning of a CL program that monitors for messages CPF0001 and CPF1999; these messages might be sent by any command processed later in the program. When either message is received from any of the commands running in the program, control branches in the program to the command identified by the label EXIT2.

CPF0001 states that an error was found in the command that is identified in the message itself. CPF1999, which can be sent by many of the debugging commands (like CHGPGMVAR), states that errors occurred on the command, but it does not identify the command in the message.

Example 2: Monitoring Messages Sent by a Single Command

```
CHGVAR VAR(&A) VALUE(&A / &B)
MONMSG MSGID(MCH1211) EXEC(CHGVAR VAR(&A) VALUE(1))
```

In this example, the MONMSG command follows a Change Variable (CHGVAR) command and, therefore, is only monitoring messages sent by the CHGVAR command. The MI escape message MCH1211 is sent to this program's message queue when a division by zero is attempted. Because MSGID(MCH1211) is specified, the MONMSG command is monitoring for this condition; when it receives the message, the second CHGVAR command is processed. In this command, the variable &A is set to a value of 1.

Error messages for MONMSG

MONSWABRM (Monitor Save While Active using BRM) Command Description

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

MONSWABRM Command syntax diagram

Purpose

The Monitor Save While Active using BRM (MONSWABRM) command allows you to review the save while active message queue and look for the message indicating the end of library synchronization. When synchronization is detected, you can issue a command to the system. The MONSWABRM command can be used as an exit (*EXIT) special value in a control group during backup processing.

Example for MONSWABRM

Example 1: Processing a Command after Synchronization

```
MONSWABRM LIB(GLLIB) CMD(SBMJOB JOB(GLDAILY))
```

In the example, the GLDAILY job is submitted when the synchronization message is sent during the save of library GLLIB.

Error messages for MONSWABRM

None

MOV (Move) Command Description

MOV Command syntax diagram

Purpose

The Move (MOV) command moves an object from the directory it is in to a different directory. If the TODIR parameter is used, the object is moved to another directory and the object keeps the same name. If the TOOBJ parameter is used the object is also renamed.

If the original object is a read-only file (a file that has the PC read-only attribute flag turned on), the MOV command operates as follows:

1. If the original file can be deleted (that is, the read-only bit can be turned off for the file), the move will succeed, retaining the read-only attribute of the file.
2. If the original file cannot be deleted, (for example, a CD-ROM file), the move operation will fail and a message will be issued indicating that the source is read-only.

When moving a file within a file system, the Last access date/time, the Data change date/time and the Attribute change date/time are preserved in the new file. If the file is moved outside of the original file system to the the "root" (/), QOpenSys, QDLS, or UDFS file systems, the Attribute change date/time is changed to the current time. >> In the case of moving to a database file member (*MBR) in the QSYS.LIB or independent ASP QSYS.LIB file system, the Data change date/time is updated as well. <<

This command can also be issued using the following alternative command name:

- MOVE

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

Restrictions:

1. The user must have object management authority for the object moved, delete and read authority to the directory the object is currently in, and add and read authority to the directory to which the object is being moved. Execute authority is required for all path prefixes.
2. Some objects are restricted and are not moved. The physical file systems return an error for these objects.
3. The directory to which the object is being moved must not already contain the name supplied in the TOOBJ parameter (or in the case where TODIR is used, the name supplied in OBJ cannot exist in TODIR).
4. Only objects that are a byte stream file type move between file systems.
5. A directory cannot be moved to a subordinate directory.
6. Database file members cannot be moved.
7. You cannot move objects in QDLS between auxiliary storage pools (ASPs).
8. >> You cannot move libraries in independent ASP QSYS.LIB to basic auxiliary storage pools (ASPs), however you can move libraries in independent ASP QSYS.LIB to the system ASP or other independent ASPs. <<

Required Parameter

OBJ Specifies the path name of the object or pattern to be moved.

The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern can be specified in the last part of the path name. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes. See path names for more information on specifying path names.

Note: An object name pattern can only be used when the TODIR parameter is used.

Optional Parameters

TODIR

Specifies the path name of the directory to which the object is being moved. The moved object uses the name specified on the OBJ parameter.

.: The path object moves to the current directory.

'directory-name': Specify the name of the directory to which the object is being moved. See path names for more information on specifying path names.

TOOBJ

Specifies the path name of the directory the object is being moved to and the new name of the object. See path names for more information on specifying path names.

Note:

The TODIR and TOOBJ parameters are mutually exclusive.

FROMCCSID

Specifies the method for obtaining the coded character set identifier (CCSID) for the source of the move operation. This CCSID will be used for data conversion, if requested. This parameter is ignored if the object specified on the OBJ parameter is not a regular file. A regular file is a file that supports the integrated file system I/O operations open, read, and write.

***OBJ:** Use the data CCSID of the object being moved.

***PCASCII:** Use the data CCSID of the object being moved to compute a CCSID in the Microsoft Windows encoding scheme (x4105). Use this as the CCSID from which the data will be converted when DTAFMT(*TEXT) is specified. This option allows data from PCs to be converted properly, if the data was created using Microsoft Windows.

***JOBCCSID:** The coded character set identifier (CCSID) from the default job CCSID is used.

from-CCSID: Specify a CCSID value between 1 and 65533.

TOCCSID

Specifies the data coded character set identifier (CCSID) for the target of the move operation. This parameter is ignored if the object specified on the OBJ parameter is not a regular file (a regular file is a file that supports the integrated file system I/O operations open, read, and write).

***OBJ:** Use the data CCSID of the object being moved. If this CCSID cannot be used by the file system that the object is being moved into, the move operation will fail.

***CALC:** Use the data CCSID of the object being moved. If this CCSID cannot be used by the file system that the object is being moved into, allow the file system to determine a different CCSID and continue with the move.

***STDASCII:** Compute a CCSID in the IBM PC Data encoding scheme (x2100), based on the source file's CCSID. Associate this CCSID with the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this CCSID for the data conversion. If this CCSID cannot be used by the file system that the object is being moved into, the move operation will fail.

***PCASCII:** Compute a CCSID in the Microsoft Windows encoding scheme (x4105), based on the source file's CCSID. Associate this CCSID with the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this CCSID for the data conversion. This option allows the resulting data to be used by Microsoft Windows applications. If this CCSID cannot be used by the file system that the object is being moved into, the move operation will fail.

***JOBCCSID:** The coded character set identifier (CCSID) from the default job CCSID is used.

to-CCSID: Specify a CCSID value between 1 and 65533. If this CCSID cannot be used by the file system that the object is being moved into, the move operation will fail.

FROMCODPAG

Specifies the method for obtaining the code page for source of the move operation. This code page will be used for data conversion, if requested. This parameter is ignored if the object specified on the OBJ parameter is not a regular file. A regular file is a file that supports the integrated file system I/O operations open, read, and write.

Note:

This keyword is replaced by FROMCCSID but the FROMCODPAG keyword can still be used. However, because this keyword may be removed in a future release, whenever possible use the FROMCCSID keyword.

***OBJ:** Use the data code page of the object being moved.

***PCASCII:** Use the data code page of the object being moved to compute a code page in the Microsoft Windows encoding scheme (x4105). Use this as the code page from which the data will be converted when DTAFMT(*TEXT) is specified. This option allows data from PCs to be converted properly, if the data was created using Microsoft Windows.

code_page: A code page value between 1-32767.

TOCODEPAGE

Specifies the data code page for the target of the move operation. This parameter is ignored if the object specified on the OBJ parameter is not a regular file (a regular file is a file that supports the integrated file system I/O operations open, read, and write).

Note:

This keyword is replaced by TOCCSID but the TOCODEPAGE keyword can still be used. However, because this keyword may be removed in a future release, whenever possible use the TOCCSID keyword.

***OBJ:** Use the data code page of the object being moved. If this code page cannot be used by the file system that the object is being moved into, the move operation will fail.

***CALC:** Use the data code page of the object being moved. If this code page cannot be used by the file system that the object is being moved into, allow the file system to determine a different code page and continue with the move.

***STDASCII:** Compute a code page in the IBM PC Data encoding scheme (x2100), based on the source file's code page. Associate this code page with the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this code page for the data conversion. If this code page cannot be used by the file system that the object is being moved into, the move operation will fail.

***PCASCII:** Compute a code page in the Microsoft Windows encoding scheme (x4105), based on the source file's code page. Associate this code page with the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this code page for the data conversion. This option allows the resulting data to be used by Microsoft Windows applications. If this code page cannot be used by the file system that the object is being moved into, the move operation will fail.

code_page: A code page value between 1-32767. If this code page cannot be used by the file system that the object is being moved into, the move operation will fail.

DTAFMT

Specifies the format of the data in the file being moved.

***BINARY:** The file contains data in binary form (such as an executable file). Do not convert data on the move. However, if the object being moved to has a different code page than the source object, all extended attributes will be converted into the code page of the new object before being set.

***TEXT:** The file contains data in textual form. Convert data to the code page of the new object during the move. The data is processed as text during the move.

If you are moving from a database member to a stream file, any line-formatting characters (such as carriage return, tab, and end-of-file) are just converted from one code page to another.

If you are moving from a stream file to a database member, the stream file must contain end-of-line characters or the move will fail. If the stream file does contain end-of-line characters, the following actions are performed during the move to a database file.

- End-of-line characters are removed.
- Records are padded with blanks (for a source physical file member) or nulls (for a data physical file member).
- Tab characters are replaced by the appropriate number of blanks to the next tab position.

Examples for MOV

```
MOV OBJ('/CURRENT/DECEMBER-1994-MONTHLY-PAYROLL-FILE')
  TODIR('/ARCHIVE')
```

This command moves a file named DECEMBER-1994-MONTHLY-PAYROLL-FILE from a directory named CURRENT to a directory named ARCHIVE.

Example 2: Moving with Conversion

```
MOV OBJ('/DATAFB')
  TOOBJ('/QSYS.LIB/APP1.LIB/DATA.FILE/DATAFB.MBR')
  TOCODEPAGE(*CALC) DTAFMT(*TEXT)
  TOCCSID(*CALC) DTAFMT(*TEXT)
```

The stream file 'DATAFB' is to be moved to the database file 'DATAFB.MBR'. By specifying TOCCSID(*CALC), the file system being moved to (the QSYS.LIB file system in this case) will try to create the new member in the same CCSID as '/DATAFB'. If this fails (in this case, if 'DATA.FILE' is not in the same CCSID as 'DATAFB'), the file system will be allowed to choose an appropriate CCSID and complete the move. By specifying DTAFMT(*TEXT), the data in 'DATAFB' is handled as text and is converted into the CCSID chosen for the new file 'DATAFB.MBR'.

Error messages for MOV

*ESCAPE Messages

CPFA085

Home directory not found for user &1.

CPFA08E

More than one name matches pattern.

CPFA093

Name matching pattern not found.

CPFA0A1

An input or output error occurred.

CPFA0A7

Path name too long.

CPFA0B0

Request not allowed to operate from one file system to another.

CPFA0B2

No objects satisfy request.

CPFA0B8

&1 objects moved. &2 objects failed.

CPFA0C4

Object name not a file.

MOVE (Move) Command Description

MOVE Command syntax diagram

MOVE Command
For the description of the MOVE command, see the Move (MOV) command description.

MOVDOC (Move Document) Command Description

MOVDOC Command syntax diagram

Purpose

The Move Document (MOVDOC) command moves a document from one folder to another, removes a document from a folder, makes a document folderless, or moves a folderless document into a folder.

Restrictions:

1. The user of this command must be enrolled in the system directory and have *ALL authority to the document.
2. To move a document to or from a folder, the user must have *CHANGE authority to the folder.
3. Documents cannot be moved between folders that reside on different auxiliary storage pools (ASPs).

Examples for MOVDOC

Example 1: Adding a Folderless Document

```
MOVDOC FROMDOC(*SYSOBJNAM) FROMFLR(*NONE) TOFLR(FLR1)
      RENAME(DOC1) SYSOBJNAM(CNTR192366)
```

This command, whose system object name is CNTR192366, adds a folderless document to FLR1 and names it DOC1.

Example 2: Moving a Document and Keeping its Name

```
MOVDOC FROMDOC(DOC1) FROMFLR(FLR1) TOFLR(FLR2)
      RENAME(*SAME)
```

This command moves DOC1 from FLR1 to FLR2 and keeps the name DOC1.

Example 3: Moving and Renaming a Document

```
MOVDOC FROMDOC(DOC1) FROMFLR(FLR1) TOFLR(FLR2)
      RENAME(DOC2)
```

This command moves DOC1 from FLR1 to FLR2 and renames it DOC2.

Example 4: Moving a Document and Making It Folderless

MOVDOC FROMDOC(DOC1) FROMFLR(FLR1) TOFLR(*NONE)

This command moves DOC1 from FLR1 and changes it to a folderless document.

Error messages for MOVDOC

*ESCAPE Messages

CPF8A13

Document &2 in folder &1 not moved.

MOVMEDBRM (Move Media Using BRM) Command Description

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

MOVMEDBRM Command syntax diagram

Purpose

The Move Media using BRM (MOVMEDBRM) command moves media based on BRMS move policies. Media that is moved as a result of processing this command must meet not only the move policy requirements (for example, allowable move day) as well as the criteria specified in the command itself for location, media class, system name and so on. The MOVMEDBRM command can be a job in the system scheduler that can run automatically or you can submit the command manually. The Volume Movement Report is produced when you run the MOVMEDBRM command. The report is written to printer file QP1AVMS.

Note: If you have a network of systems that use BRMS, it is only necessary to process the MOVMEDBRM command on one of the members of the network, although the process can be done on a system by system basis.

Note: The system in the network that is running the movement for all of the other systems in the network should be physically attached to all media libraries that support the network operations. If this is not the case, you may have to run MOVMEDBRM again, specifying the appropriate move policy for the logically attached media library device.

Example for MOVMEDBRM

Example 1: Selecting all volumes to move for location *HOME

```
MOVMEDBRM LOC(*HOME)
```

In the example, all volumes for all move policies that are located at the location *HOME, are selected for media movement.

Error messages for MOVMEDBRM

None

MOVOBJ (Move Object) Command Description

MOVOBJ Command syntax diagram

Purpose

The Move Object (MOV OBJ) command removes an object from its currently assigned library and places it in a different library. The type of the object moved is specified in the OBJTYPE parameter.

When this command is used to move an object, the save and restore information is removed from the object description.

Restrictions:

1. The user who submits this command must have object management authority for the object moved, delete and execute authority to the library the object is currently in, and add and read authority to the library to which the object is being moved.
2. >> The user must have *USE authority to the auxiliary storage pool (ASP) device if a specific ASP device name is specified on the ASPDEV or TOASPDEV parameter.<<<
3. >> Libraries, user profiles, edit descriptions, line descriptions, control unit descriptions, and device descriptions cannot be moved.<<<
4. >> Journals and journal receivers can be only be moved from a reclaim storage library (QRCL or QRCLxxxxx, where 'xxxxx' is the number of a primary ASP) to the library where the journal object was originally created.<<<
5. The following objects cannot be moved: the system operator message queue QSYSOPR, all work station user message queues, and the system log QHST.
6. The library to which the object is being moved must not already contain an object of the same name and type as the object being moved.
7. >> The library to which the object is being moved cannot be QTEMP.<<<
8. The user space (*USRSPC), user index (*USRIDX), and user queue (*USRQ) user domain objects can only be moved into libraries that are permitted in the system value QALWUSRDMN (allow user objects in library). However, if the user object was created in the system domain, it is not restricted.
9. >> As a general rule, objects cannot be moved to a library if the object and the library are in different auxiliary storage pools (ASPs). An error message is issued when the object cannot be moved. There are some specific exceptions to the general rule:
 - You can move save files that are in a basic user ASP to libraries that are in the system ASP (ASP 1) if the save file's library is also in the system ASP.
 - You can move objects in a secondary ASP to the primary ASP in the same ASP group if the target library is QRPLxxxxx (where 'xxxxx' is the number of the primary ASP of the ASP group.)
 - You can move an object from the QTEMP library (always in the system ASP) to a library in a primary or secondary ASP with the following considerations:
 - a. The object type is valid for a primary or secondary ASP.
 - b. The 'move' is accomplished through a save and restore operation. You must have authority to the CRTSAVF command.
 - c. The size of the object must be less than 1 terabyte. The Move Library to ASP (QHSMMOV L) API does not have this size limitation. To use the MOVE Library to ASP (QHSMMOV L) API, you would first need to use the MOV OBJ command to move the object from QTEMP to another library in the system ASP (ASP 1).
 - d. For data queues, message queues, and logical files, only the object descriptions are moved. The contents of the objects are not moved.
 - e. After the object has been moved, the following attributes will differ from the original object:
 - The date last used will be set to blank.
 - The change date and time will be set to the current date and time.
 - The days used count will be set to zero.
 - The date use count reset will be set to blank.
 - The save and restore information will be set to blank.

- f. The private authorities to the object will be preserved. <<

>> Required Parameters

OBJ Specifies the qualified name of the object being moved to another library. (If no library qualifier is given, *LIBL is used to find the object.) The object name should be qualified to ensure that the correct object is moved.

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 that is moved.

OBJTYPE

Specifies the type of the object moved to another library. More information on this parameter is in Commonly used parameters.

TOLIB Specifies the name of the library to which the object is moved. Library QTEMP cannot be specified.

***CURLIB:** The object is moved to the current library. 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 which the object is moved.

Optional Parameters

ASPDEV

Specifies the auxiliary storage pool (ASP) device name where storage is allocated for the library containing the object to be moved (OBJ parameter). If the library resides in an ASP that is not part of the library name space associated with the thread, this parameter must be specified to ensure the correct object is moved. If this parameter is used when *LIBL or *CURLIB is specified for the OBJ parameter, * is the only valid value.

***:** The ASPs that are currently part of the thread's library name space will be searched to locate the library. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and, if the thread has an ASP group, the primary and secondary ASPs in the ASP group.

***CURASPGRP:** If the thread has an ASP group, the primary and secondary ASPs in the ASP group will be searched to locate the library. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched.

***SYSBAS:** The system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) will be searched to locate the library. No primary or secondary ASPs will be searched, even if the thread has an ASP group.

auxiliary-storage-pool-device-name: The device name of the primary or secondary ASP to be searched to locate the library. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Available'. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched.

TOASPDEV

Specifies the auxiliary storage pool (ASP) device name where storage is allocated for the library where the object is to be moved (TOLIB parameter). If the library is in an ASP that is not part of the thread's library name space, this parameter must be specified to ensure the object is moved to the correct library. If this parameter is used when *CURLIB is specified for the TOLIB parameter, either * must be specified or *ASPDEV must be specified and the ASPDEV parameter must be *.

***ASPDEV**: The ASP device specified for the ASPDEV parameter will be searched to locate the library.

*****: The ASPs that are currently part of the thread's library name space will be searched to locate the library. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and, if the thread has an ASP group, the primary and secondary ASPs in the ASP group.

***CURASPGRP**: If the thread has an ASP group, the primary and secondary ASPs in the ASP group will be searched to locate the library. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched.

***SYSBAS**: The system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) will be searched to locate the library. No primary or secondary ASPs will be searched, even if the thread has an ASP group.

auxiliary-storage-pool-device-name: The device name of the primary or secondary ASP to be searched to locate the library. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Available'. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched. <<

Examples for MOVOBJ

Example 1: Moving an Object from the General Purpose Library

```
MOVOBJ OBJ(QGPL/X) OBJTYPE(*PGM) TOLIB(MY)
```

The general purpose library is searched for the program (*PGM) object X. Before X is moved to MY library, the user profile of the user submitting the command is checked for (1) object operational and management authority for the object, (2) add and execute authority for MY library, and (3) delete and read authority for the library from which the object X is moved. After this command is run, object X is no longer in the QGPL library.

Example 2: Moving an Object from a Library in the Library List

```
MOVOBJ OBJ(*LIBL/Y) OBJTYPE(*FILE) TOLIB(Z)
```

or

```
MOVOBJ Y *FILE Z
```

The library list (*LIBL) is used to locate the file named Y. >> If more than one file object with the same name exists in the libraries making up the library list, the first object found in the library list is moved. <<

Error messages for MOVOBJ

*ESCAPE Messages

CPFA030

Object already in use.

» CPFB8ED

Device description &1 not correct for operation. «

CPF0601

Not allowed to do operation to file &1 in &2.

CPF0602

File &1 already in library &2.

CPF0605

Device file &1 in &2 saved with storage freed.

CPF0610

File &1 in &2 not available.

CPF0678

Operation not performed for file name &1 in &2.

CPF1763

Cannot allocate one or more libraries.

CPF2105

Object &1 in &2 type *&3 not found.

CPF2110

Library &1 not found.

CPF2112

Object &1 in &2 type *&3 already exists.

CPF2113

Cannot allocate library &1.

CPF2114

Cannot allocate object &1 in &2 type *&3.

CPF2135

Object &1 type *&3 already exists in library.

CPF2150

Object information function failed.

CPF2151

Operation failed for &2 in &1 type *&3.

CPF2160

Object type *&1 not eligible for requested function.

» CPF216C

TOASPDEV value not allowed with TOLIB(*CURLIB). «

» CPF2173

Value for ASPDEV not valid with special value for library. «

» CPF218C

&1 not a primary or secondary ASP. «

CPF2182

Not authorized to library &1.

CPF2183

Object &1 cannot be moved into library &3.

CPF2189

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

CPF2193
Object &1 cannot be moved into library &4.

CPF22BC
Object &1 type &3 is not program defined.

» **CPF2451**
Message queue &1 is allocated to another job. «

CPF2512
Operation not allowed for message queue &1.

CPF32CF
Distributed file error, reason code &3.

CPF32C3
Distributed file error, level ID mismatch

CPF320B
Operation was not valid for database file &1.

CPF320C
File &1 not allowed in SQL collection &2.

CPF3201
File &1 in library &2 already exists.

CPF3202
File &1 in library &2 in use.

CPF3203
Cannot allocate object for file &1 in &2.

CPF322D
Operation not done for data base file &1.

CPF3220
Cannot do operation on file &1 in &2.

» **CPF3224**
Not authorized to perform operation on file &1. «

CPF323C
QRECOVERY library could not be allocated.

CPF323D
User does not have correct authority.

» **CPF323F**
Move or rename of file &1 in library &2 not complete. «

CPF3231
Cannot move file &1 from library &2.

CPF324B
Cannot allocate dictionary for file &1.

CPF324C
Concurrent authority holder operation prevents move, rename or restore.

CPF3245
Damage to file &1 member &6 prevents operation on file &3.

CPF325D
Field CCSID values not compatible.

CPF327C

File &1 cannot be moved into library &4.

CPF327E

Alternative name for file &1 not allowed.

» CPF329D

Operation not successful for file &1 in library &2. «

CPF3323

Job queue &1 in &2 already exists.

CPF3330

Necessary resource not available.

CPF3353

Output queue &1 in &2 already exists.

CPF3373

Job queue &1 in &2 not moved. Job queue in use.

CPF3374

Output queue &1 in &2 not moved. Output queue in use.

CPF3467

Output queue &1 deleted and then created again.

CPF3469

Operation not allowed for output queue.

CPF7003

Entry not journaled to journal &1. Reason code &3.

CPF7010

Object &1 in &2 type *&3 already exists.

CPF7014

Object &1 cannot be moved to library &4.

CPF9807

One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

» CPF9814

Device &1 not found. «

» CPF9825

Not authorized to device &1. «

CPF9827

Object &1 cannot be created or moved into &2.

» CPF9833

*CURASGRP or *ASPGRPPRI specified and thread has no ASP group. «

» CPF9876

Protected library &2 cannot be modified. «

» CPF9899

Error occurred during processing of command. «

»

MOVSPFBRM (Move Saved Spooled Files Using BRM) Command Description

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

MOVSPFBRM Command syntax diagram

Purpose

The Move Saved Spooled Files using BRM (MOVSPFBRM) command provides for the movement of selected spooled files to a specified library-qualified output queue. Selection criteria include library-qualified from output queue name, from ASP, spooled file, job name, job user, job number, user data, create date range, last used date range, and size range. A run option of *REPORT is provided to allow the user to review the Move Spooled Files using BRM report prior to moving the selected spooled files. The report, if printed, is written to printer file QP1AMSF.

Using the ASP attribute of the spooled file for the output queue, BRMS determines whether spooled files are actually moved from one ASP to another ASP when moved from one output queue to another. If a spooled file that moves from one output queue to another does not move from one ASP to another, then BRMS does not check the ASP high storage threshold of the ASP because the spooled file does not move. If a spooled file that moves from one output queue to another also moves from one ASP to another, then before requesting the spooled file to move, BRMS determines if the target ASP has sufficient space to accommodate the spooled file without exceeding the high storage threshold of the target ASP. If the spooled file cannot be moved without exceeding this threshold, BRMS shows that the file was not moved. The file is included in the summary section detail that indicates the number of files and amount of spooled data that could not be moved.

Restriction: The Advanced Functions feature is required to use this command.

Example for MOVSPFBRM

Example 1: Move Large Spooled Files in System ASP

```
MOVSPFBRM OPTION(*MOVE) TOOUTQ(MYLIB/MYOUTQ)
FROMASP(*SYSTEM) SLTSIZE(*MB 50 *NOMAX)
```

In this example you are specifying that all spooled files currently in the system ASP that are fifty or more megabytes in size should be moved to the output queue named MYOUTQ in the library named MYLIB. This example assumes MYLIB is not in the system ASP, and the ASP attribute of the spooled file for the MYOUTQ output queue specifies that spooled files are placed in the same ASP as the output queue.

Error messages for MOVSPFBRM

No error messages. <<

NETSTAT Command

NETSTAT Command

For a description of the NETSTAT command, see the WRKTCPSTS (Work with TCP/IP Network Status) command description.

WRKTCPSTS syntax diagram

OPNQRYP (Open Query File) Command Description

OPNQRYP Command syntax diagram

Purpose

The Open Query File (OPNQRYP) command opens a file that contains a set of database records that satisfies a database query request. Once opened, the file looks like a database file opened by using the Open Database File (OPNDBF) command, and the records in the file are accessed by high-level language programs that share the open data path (ODP). The path is closed, and all query resources are deallocated, by using the Close File (CLOF) command.

This command is used to do any combination of the following database functions:

- Join records from more than one file, member, and record format. The join operation that is performed may be equal or non-equal in nature.
- Calculate new field values by using numeric and character operations on field values and constants.
- Group records by like values of one or more fields, and calculate aggregate functions, such as minimum field value and average field value, for each group.
- Select a subset of the available records. Selection can be done both before and after grouping the records.
- Arrange result records by the value of one or more key fields.

More Command Functions: Following the parameter descriptions and the command examples in the Additional Considerations section. Included in it are subsections on:

- Field Names
- Expressions
- Built-In Functions
- Restricted Built-In Functions

Restrictions:

1. The user can use overrides to change the file, library, and member names specified on the FILE parameter. Overrides are ignored for the file and library specified on the FORMAT parameter, unless FORMAT(*FILE) is specified. Parameter values specified on an override command, other than TOFILE, MBR, LVLCHK, WAITRCD, SEQONLY, or INHWRT and SHARE, are ignored by the OPNQRYP command.
2. The OPNQRYP command does not share an existing open data path (ODP) in the job or activation group. If an existing SHARE(*YES) ODP in the job or activation group has the same file, library, and member name as the open query file open data path (ODP), the query file does not open and an escape message is sent.
3. Each subsequent shared open operation must use the same open options (such as SEQONLY) that are in effect when the OPNQRYP command is run.
4. Some system functions (such as the Display Physical File Member (DSPPFM) and Copy File (CPYF) commands) do not share an existing open data path. The OPNQRYP command cannot be used with those functions.
5. The file opened with the OPNQRYP command cannot be used in programs written in BASIC because BASIC does not share an existing open data path.
6. In multithreaded jobs, this command is not threadsafe for distribute 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.
7. Users of this command must have the following authorities:
 - Execute authority for any library that is needed to locate the files specified on the FILE and FORMAT parameters

- Object operational authority and one or more of the following data authorities for any physical file or logical file specified on the FILE parameter:
 - Read authority if the file is opened for input (using option *INP)
 - Add authority if the file is opened for output (using option *OUT)
 - Update authority if the file is opened for updates (using option *UPD)
 - Delete authority if the file is opened for deletions (using option *DLT)
 - Read, add, update, and delete authority if the file is opened for all I/O operations (using option *ALL)
- Object operational authority for any file specified on the FORMAT parameter
- Use authority for any translate tables specified on the MAPFLD parameter (using option *USE)

Notes:

1. The Copy from Query File (CPYFRMQRYF) command can be used to copy data from a data path opened with the OPNQRYF command.
2. More information about the OPNQRYF command is in the Database Programming topic in the Information Center.

Required Parameter

FILE Specifies one or more files, members, and record formats that are processed by the open query file command. All files specified must be physical files, logical database files, or Distributed Data Management (DDM) files. If DDM files are used, all files they refer to must be on the same target system, and the target system must be an IBM iSeries 400 computer or IBM System/38.

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 files, members, and record formats that are processed.

Element 1: Member Values

***FIRST:** The oldest member created is used.

***LAST:** The newest member created is used.

member: Specify the file member name.

Element 2: Record Format Values

***ONLY:** Only the record format in the file is used. If no record format name is specified, *ONLY is the default. If the file has more than one record format, a record format name is specified.

record-format-name: Specify the name of the record format that is used. The record format must exist in the database file specified in the first part of the FILE parameter.

When more than one file, file member, and record format is specified, the query joins field values to create a single set of records. Any file specified in the list may be a physical, join logical, or SQL view. More information on SQL views is in the SQL Reference topic in the Information Center. The maximum number of physical file members that can be joined by a single query is 32. This limit includes the based-on physical file members for any join logical file member or SQL view member specified on the FILE parameter. The limit also includes each separate occurrence of the same physical file member when it is specified more than once in the list, either directly by name or by being referred to through a logical file member.

Optional Parameters

OPTION

Specifies the open option used for the query file. The options chosen on the first full open operation of a file are not changed on subsequent shared open operations. The user can specify either OPTION(*ALL) or up to four of the following values in any order:

***INP:** Open the file for input. OPTION(*INP) is the only value allowed if join processing or group processing is requested, if UNIQUEKEY processing is specified, or if all the fields in the open query file record format (specified on the FORMAT parameter) are for input-only use.

***OUT:** Open the file for output. In some high-level languages, output to certain files (such as files defined as 'direct access' in the high-level language program) is done by using a combination of input and updates. OPTION(*UPD) or OPTION(*ALL) is specified to use an open query file with such a program.

***UPD:** Open the file for update operations. If an input operation comes before an update, *INP on the OPTION parameter must be specified when OPTION(*UPD) is specified.

***DLT:** Open the file for delete operations. If each delete operation is preceded by an input operation, *INP on the OPTION parameter must be specified when OPTION(*DLT) is specified.

Other Single Values

***ALL:** Opens the file for all operations (*INP, *OUT, *UPD, and *DLT).

FORMAT

Specifies the record format used for records made available by using the OPNQRYF command. The simple field names in the open query file record format must represent fields that are either defined on the MAPFLD parameter or are unique across all files, members, and record formats specified on the FILE parameter. The value for any field that has the same name as a field specified on the MAPFLD parameter is determined by the mapped-field definition on the MAPFLD parameter. The value for any field not defined on the MAPFLD parameter is determined by a mapping of the field with the same name in one of the based-on files, file members, and record formats specified on the FILE parameter. Only the name, type, length, decimal positions, keyboard shift, and usage attributes of each field specified in the record format that is identified on the FORMAT parameter are used for processing the OPNQRYF command. All other attributes are ignored. The attributes do not have to be the same. If they differ, the fields are mapped in a way similar to those described for the Change Variable (CHGVAR) command.

***FILE:** The record format of the first or only entry on the FILE parameter is used. FORMAT(*FILE) is not allowed when more than one file, member, and record format are specified on the FILE parameter (therefore requiring a join query).

Element 1: File Name

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

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

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

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

database-file-name: Specify the name of a physical or logical database file, or a Distributed Data Management (DDM) file that contains the record format that is used.

Element 2: Record Format Values

***ONLY:** The only record format in the file is used. If no record format name is specified, *ONLY is the default. If the file has more than one record format, specification of a record format name is required.

record-format-name: Specify the name of the record format that is used. The record format must exist in the database file specified in the first part of the FORMAT parameter.

QRYSLT

Specifies the selection values used (before grouping) to determine which records are available by using the open query file.

***ALL:** All records in the physical or logical files, members, and record formats specified on the FILE parameter (after join processing, if required) are selected.

'query-selection': Specify an expression of up to 5000 characters (enclosed in apostrophes) that describes the values used to determine which records are selected. Any logical expression formed from relations can be specified (such as *EQ and *NE) of field and constant values or functions of field and constant values. At least one field name is specified in each relation. However, a field cannot be specified that depends on an aggregate function either directly in its definition or indirectly by referring to a mapped field. Refer to the "Expressions" section of Additional Considerations for a complete list of the operators used for this parameter.

For example, to select all records for which the value of field CUSNBR is less than 7000 and the value of field BALDUE is greater than 90% of the value of field CRLIMIT, specify the following:

```
QRYSLT ('CUSNBR<7000 *AND  
BALDUE>CRLIMIT*0.9')
```

Each field name may be qualified with either a file name or number that indicates which element in the list of files, members, and record formats specified on the FILE parameter contains the field. The special value *MAPFLD may be used to qualify the field name if the field is defined on the MAPFLD parameter.

KEYFLD

Specifies the name of one or more key fields used to arrange the query records, or specifies that the access path sequence of the first or only file, file member, and record format specified on the FILE parameter is used to arrange the query records. If key field names are specified, the user must also indicate whether the part of the key associated with each key field is ascending or descending, and whether the records are arranged by the absolute value of a numeric key field. If the qualified key field specified is a double-byte character field, the data is arranged in a single-byte sequence.

All key fields for an open query file must appear in the query records processed through the file. The fields named in the record format identified on the FORMAT parameter must include all key fields for the open query file, even if KEYFLD(*FILE) is specified to use the existing access path of a keyed file.

***NONE:** Key fields are not used to arrange the query records; therefore, any arrangement is acceptable. It is possible for the system to display query records in different arrangements if the same query is run twice, based on such factors as the current number of records in the file members being queried. KEYFLD(*NONE) allows the system more flexibility than other KEYFLD values to improve the performance record processing through the open query file.

***FILE:** The query records have the same arrangement as the first file, file member, and record format specified on the FILE parameter. KEYFLD(*FILE) is specified even if the first file in the list has only an arrival sequence access path, in which case the query record arrangement matches the arrival sequence of the first file, file member, and record format specified on the FILE parameter.

When KEYFLD(*FILE) is specified and a sort sequence other than *HEX has been specified on the SRTSEQ parameter, you may receive your records in an order that does not reflect the true file order. If this is a key file, the query's sort sequence is applied to the key fields of the file and an informational message is sent. If the file has a sort sequence table or an alternative collating sequence table, it is ignored for the ordering. This allows users to indicate which fields to apply a sort sequence to without having to list all the field names. If a sort sequence is not specified for the query, the query is ordered as in releases previous to V2R3M0.

Element 1: Qualified Key Field Values

qualified-key-field-name: Specify one or more field names (up to 50 field names can be entered) used to define a keyed access path to arrange the query records. Each field name is qualified with either a file name or number that indicates which element in the list of files, members, and record formats specified on the FILE parameter contains the field. The special value *MAPFLD is also used to qualify the field name if the field is defined on the MAPFLD parameter.

Each field name on the KEYFLD parameter must be a field name that is defined in the query records specified on the FORMAT parameter. For example, if the value *MAPFLD is specified on the KEYFLD parameter, the name of the mapped field must be defined in the query records specified on the FORMAT parameter. The sum of the lengths of all key fields cannot be more than 10,000 bytes. In addition, if the sum of the lengths of the key fields is greater than 2000 bytes, *INP must be specified on the OPTION parameter.

Element 2: Key Field Order

***ASCEND:** The part of the key defined by the specified key field is ordered by ascending key values.

***DESCEND:** The part of the key defined by the specified key field is ordered by descending key values.

Element 3: Order by Absolute Value

***ABSVAL:** The part of the key defined by the specified key field is arranged by the absolute value of the key field. *ABSVAL is specified together with either *ASCEND or *DESCEND, but it is ignored if the key field is not numeric. If *ABSVAL is not specified, the records are arranged by the signed value of a numeric key field.

UNIQUEKEY

Specifies whether the query is restricted to records with unique key values, and specifies how many of the key fields must be unique. If *ALL or *number-of-key-fields* is specified, null values are considered to be equal.

***NONE:** None of the key fields specified on the KEYFLD parameter must be unique. All query records are available through the open query file, regardless of key value.

***ALL:** All key fields specified on the KEYFLD parameter must be unique. If there are multiple query records with the same values for all of the key fields, only the first such record is available through the open query file.

number-of-key-fields: Specify the number of key fields that must be unique. This value must be no larger than the number of key fields determined by the KEYFLD parameter. If there are multiple query records with the same value for the specified number of consecutive key fields, only the first such record is available through the open query file.

JFLD Specifies whether the query joins records from multiple file members, and specifies how to join field values from the files, members, and record formats specified on the FILE parameter in constructing the query records.

The first file, member, and record format specified on the FILE parameter is called the join primary, and all other elements specified on the FILE parameter are called join secondaries. The JFLD parameter specifies a list containing pairs of field names, where the first field in each pair provides a value that is used to select records in a join secondary that has the same value in the second field name of the pair.

The join from-field and to-field may be simple or mapped fields (specified on the MAPFLD parameter), but a field that depends on an aggregate function (either directly in its definition or indirectly by referring to a mapped field) cannot be used.

The join from-field and to-field are not required to have identical field attributes, but numeric fields cannot be mixed with character or double-byte character set (DBCS) fields, and character fields cannot be mixed with DBCS-only field types. If the fields do not have identical attributes, the system converts them to identical attributes for join processing. This change uses a packed decimal format if both fields are fixed-point numeric fields, or floating-point format if either field is of the floating-point type. The change for fixed-point numeric fields aligns the decimal points and pads with zeros. Numeric-type change truncates fractional digits if more than 31 total digits are required for fixed-point numbers, or drops some of the least significant digits if more than 15 total digits are required for floating-point numbers. Character and DBCS-open fields are changed by padding the shorter field with blanks. DBCS-either fields are padded with blanks if the field contains SBCS data or padded with double-byte blanks if the field contains DBCS data. DBCS-only fields are padded with double-byte blanks.

If more than one file is specified on the FILE parameter, and if the JDFTVAL(*NO) value and JORDER(*ANY) value are specified, the system takes information from the JFLD and QRYSLT parameters and derives the final join specifications. If a file is specified on the FILE parameter and the file is not referred to by the QRYSLT or JFLD parameters, all records for that file are logically joined to all other records created from the other files specified on the FILE parameter.

If either JDFTVAL(*YES), JDFTVAL(*ONLYDFT), or JORDER(*FILE) is specified, the join fields must be specified on the JFLD parameter.

***NONE**: No join operation is specified. If more than one file is specified on the FILE parameter, and JDFTVAL(*NO) and JORDER(*ANY) is specified, the system automatically finds the join fields from the QRYSLT parameter.

Element 1: From-Field Name

qualified-from-field-name: Specify a field name to provide the value used to select records in a join secondary file, file member, and record format. The field name is qualified with either a file name or number that indicates which element in the list of files, file members, and record formats (specified on the FILE parameter) contains the field. The special value *MAPFLD can also be used to qualify the field name if the field is defined on the MAPFLD parameter.

A join from-field is either a simple field or a mapped field defined on the MAPFLD parameter. If JDFTVAL(*YES) or JDFTVAL(*ONLYDFT) is specified, it must depend only on fields contained in the join primary or in join secondaries specified on the FILE parameter ahead of the join secondary associated with the to-field of the pair.

Element 2: To-Field Name

qualified-to-field-name: Specify a field name used to select records from a join secondary file, file member, and record format in constructing the query records. The field name is qualified with

either a file name or number that indicates which element in the list of files, file members, and record formats specified on the FILE parameter contains the field. The special value *MAPFLD is used to qualify the field name if the field is defined on the MAPFLD parameter.

A join to-field is either a simple field or a mapped field defined on the MAPFLD parameter. If JDFTVAL(*YES) or JDFTVAL(*ONLYDFT) is specified, it must depend only on fields contained in a single join secondary. If the join secondary is a join logical file, only fields contained in the primary physical file member for the join logical file are components of the join to-field. The sum of the lengths of all to-fields for each join secondary (after change, if the from-field and to-field attributes are not identical) cannot be more than 2000 bytes unless JDFTVAL(*NO) is specified. Up to 50 join field pairs are specified.

Element 3: Join Operators

join-operator: Specifies the type of join operation that is performed for the specified from-field and to-field. If JDFTVAL(*NO) and JORDER(*ANY) are specified, or if more than one join field pair is specified, a different join operator may be specified for each pair. If JDFTVAL(*YES), JDFTVAL(*ONLYDFT), or JORDER(*FILE) is specified, then only one join operator may be specified, regardless of the join pairs.

***EQ**: An equal join operation is performed.

***GT**: A greater than join operation is performed.

***LT**: A less than join operation is performed.

***NE**: A not equal join operation is performed.

***GE**: A greater than or equal join operation is performed.

***LE**: A less than or equal join operation is performed.

JDFTVAL

Specifies whether the query file includes join records that use default values for any of the fields. The default values are from a join secondary file that contains a record that does not match the corresponding record in the primary file. The matching attempted is the join criteria as specified on the JFLD parameter.

Join processing attempts to collect field values from the join primary and join secondaries. It does so by matching join from-field values to records in a join secondary that produce the appropriate values in the join to-field. If there are no records in a join secondary to produce the to-field values required for the pairs of join fields associated with the join secondary, this parameter specifies whether query records should be constructed by using default values for all fields obtained from the join secondary.

If the FILE parameter includes any join logical or SQL views, all join files must be compatible with the JDFTVAL value. If the data description specification (DDS) used to create a queried join logical file does not contain the JDFTVAL keyword, JDFTVAL(*NO) must be specified. If any join logical file specified on the FILE parameter has the JDFTVAL keyword, then all join logical files for this open query file must be created by using the JDFTVAL keyword, and JDFTVAL(*YES) must be used. If any files are SQL views, then JDFTVAL(*NO) is required.

If the JDFTVAL parameter is not compatible with the attributes of the join logical files or SQL views being processed, the join view files specified on the FILE parameter can be replaced with their based-on physical file members. The correct, additional from-field and to-field pairs can be provided on the JFLD parameter to join records from the physical file members in any way.

If more than one file is specified on the FILE parameter, and if either JDFTVAL(*YES) or JDFTVAL(*ONLYDFT) is specified, the system uses the join fields as specified on the JFLD parameter as the final join specifications.

***NO**: Default values are not used to construct join query records.

***YES:** Creates all records for the join operation, including those produced both with and without default values. If *YES is specified, no SQL views are allowed.

***ONLYDFT:** Creates only the records produced by using default values in constructing the join operation. This option is used to include only exception records in the records available through the open query file. If *ONLYDFT is specified, no join logical files or SQL views can be specified on the FILE parameter.

JORDER

Specifies, for a join query, whether the join order must match the order specified on the FILE parameter. If the join order is varied, the query records are created in a different arrangement. If the value specified for the JDFTVAL parameter is *YES or *ONLYDFT, the JORDER parameter is ignored. The order specified on the FILE parameter is always preserved, because changing the join order can change which records are returned when JDFTVAL processing is required.

If more than one file is specified in the FILE parameter and JORDER(*FILE) is specified, the system uses the join fields as specified on the JFLD parameter as the final join specifications.

***ANY:** Any join file order is allowed, and any such arrangement may be used by the system to create the result records. It is possible for a query to return result records in a different arrangement if the same query is run twice, consecutively (based on factors such as the current number of records in the files being queried). JORDER(*ANY) allows the system more flexibility to improve the performance of processing records through the open query file than any other JORDER parameter value.

***FILE:** The order of the file, file member, and record format elements specified on the FILE parameter are preserved in the join operation.

GRPFLD

Specifies the field name or names used to group query results. One query record is created for each group of records (after join processing, if required) selected by the QRYSLT parameter. The group is defined by the collection of records that has the same set of values for the fields specified in the record format identified on the FORMAT parameter. If no field names are specified and group processing is required, the whole file is considered to be one group. Each query record that is created is either made available through the open query file or is discarded, depending on the selection values specified on the GRPSLT parameter. All null values within a grouping column are considered equal. To ensure an ascending sequence, the KEYFLD parameter must also be specified.

***NONE:** Fields are not used to form groups. If the grouping function is required (because selection values are specified on the GRPSLT parameter, or an aggregate function is used by a field specified on the MAPFLD parameter), records selected by the values specified on the QRYSLT parameter are handled as a single group.

qualified-group-field: Specify one or more field names (up to 50) to group the query results. Each field name may be qualified with either a file name or number to indicate which element in the list of files, members, and record formats specified on the FILE parameter contains the field. The special value *MAPFLD may also be used to qualify the field name if the field is defined on the MAPFLD parameter.

A grouping field defined on the MAPFLD parameter cannot refer to an aggregate function in its definition (either directly, or indirectly through the use of another field specified on the MAPFLD parameter). The sum of the lengths of all grouping fields cannot exceed 2000 bytes.

GRPSLT

Specifies the selection values used after grouping to determine which records are available through the open query file.

***ALL:** All records defined by the grouping function described in the GRPFLD parameter description are selected.

'group-selection': Specify an expression (up to 2000 characters enclosed in apostrophes) that describes the values used to determine which records are selected. Any logical expression formed from relations (such as *EQ and *NE) of field and constant values or functions of field and constant values are specified. Only grouping fields (specified on the GRPFLD parameter), constants, aggregate functions (such as %AVG and %STDDEV), and mapped fields (specified on the MAPFLD parameter) that are composed of grouping fields and aggregate functions can be referred to in any relation. At least one field must be specified in each relation. Refer to the "Expressions" section of Additional Considerations for a complete list of the operators used for this parameter.

For example, to select all records for which the value of grouping field OVRDUE is greater than or equal to 10, and the average value of field CREDIT in each group is less than 100, specify the following:

```
GRPSLT('OVRDUE>=10 *AND %AVG(CREDIT)<100')
```

Each field name may be qualified with either a file name or number that indicates which element in the list of files, file members, and record formats specified on the FILE parameter contains the field. The special value, *MAPFLD, may also be used to qualify the field name if the field is defined on the MAPFLD parameter.

MAPFLD

Specifies the definition of query fields that are either mapped to, or derived from, other fields. MAPFLD is generally not needed if the field names specified on other OPNQRYF command parameters are simple field names that exist in only one of the file, file member, and record format elements specified on the FILE parameter.

***NONE:** Mapped fields are not needed. All field names specified on other parameters exist in some record format specified on the FILE parameter.

Element 1: Mapped Field

mapped-field-name: Specify the simple field name used on any other OPNQRYF command parameter that must refer to this mapped field. A qualified name is *not* allowed for the first part of the MAPFLD parameter list item. All specified mapped-field-name values must be unique. Refer to the "Expressions" section of Additional Considerations for a complete list of the operators used for this parameter.

Element 2: Field Definition Expression

'mapped-field-definition': Specify an expression (up to 256 characters enclosed in apostrophes) that defines the mapped field in terms of other fields that either exist in only one of the file, file member, and record format elements specified on the FILE parameter, or are defined by some other mapped field definition appearing earlier in the MAPFLD list. Either numeric operations or string operations are allowed, depending on the data type of the fields used in the definition. Refer to the "Expressions" section of Additional Considerations for a complete list of the operators used for this parameter.

For example, to define a mapped field named WHSPAR as the concatenation of the field WHRSE with the first 10 characters of field PART, specify the following:

```
MAPFLD((WHSPAR 'WHRSE *CAT %SST(PART 1 10)'))
```

Each field name is qualified with either a file name or a number that indicates which element in the list of files, file members, and record formats specified on the FILE parameter contains the field. The special value *MAPFLD is also used to qualify the field name if the field is defined in an earlier list item on the MAPFLD parameter.

Element 3: Mapped Field Type

field-type: Specify the field type for this mapped field, or specify *CALC to allow the system to calculate appropriate attributes (including field type) for the mapped field. *CALC is the default if no field type value is specified.

When *CALC is used, the field attributes are determined in one of two ways. The attributes either match the field definition in the record format identified on the FORMAT parameter, or (if the field is not in the record format on the FORMAT parameter) the attributes are calculated based on the expression specified in the mapped-field definition for this field. If the mapped field is used in the record format identified on the FORMAT parameter, either use *CALC or specify attributes (field type, field length, and field decimals) identical to those of the field in the record format specified on the FORMAT parameter.

The field type must be valid for the final result of the expression specified on the mapped-field definition. Character and numeric types are only mixed if the numeric type is in zoned decimal format and the field length and the length of the expression result are the same.

The following are the only DBCS mappings allowed on the MAPFLD parameter:

- From character to DBCS-open type
- From character to DBCS-either type
- From character to DBCS-graphic type
- From character to UCS2 graphic type
- From HEX to DBCS-open type
- From HEX to DBCS-either type
- From HEX to DBCS-only type
- From HEX to DBCS-graphic type
- From HEX to UCS2-graphic type
- From DBCS-either to DBCS-open type
- From DBCS-either to HEX type
- From DBCS-either to DBCS-graphic type
- From DBCS-either to DBCS-either type
- From DBCS-open to DBCS-open type
- From DBCS-open to character type
- From DBCS-open to DBCS-graphic type
- From DBCS-open to HEX type
- From DBCS-open to UCS2 graphic type
- From DBCS-only to DBCS-only type
- From DBCS-only to DBCS-open type
- From DBCS-only to DBCS-either type
- From DBCS-only to DBCS-graphic type
- From DBCS-only to HEX type
- From DBCS-graphic to DBCS-only type
- From DBCS-graphic to DBCS-open type
- From DBCS-graphic to DBCS-either type
- From DBCS-graphic to DBCS-graphic type
- From DBCS-graphic to UCS2-graphic type
- From UCS2-graphic to HEX type
- From UCS2-graphic to DBCS-open type
- From UCS2-graphic to character type

- From UCS2-graphic to DBCS-graphic type
- From UCS2-graphic to UCS2-graphic type

The Query Field Structure table (see Table 1 (page 35)) shows the allowable *field type* and *external field length* values. It also shows the *default field length* and *decimal positions* and the *internal field length* (in bytes) for each type of field.

Element 4: Length

field-length: Specify the field length in number of digits for a numeric field, number of bytes for a character or DBCS field, or number of characters for a graphic field. A field length must be an even value for DBCS-only and DBCS-either field types. The range of valid lengths for each field type is shown in the Query Field Structure table. A field length value is specified if *CALC is used for the field type.

Element 5: Decimal Positions

field-decimals: Specify the number of decimal positions for a numeric field, expressed as a number of decimal digits, that is no larger than the total number of digits specified for the field length. If no value is given, the default value is assumed to be zero. A field decimal's value must not be specified for a binary or character field, or if *CALC is specified for the field type.

Element 6: CCSID Value

field-CCSID: Specify the CCSID (character code set identifier) being assigned to the mapped field being created. This parameter is valid only for fixed- or variable-length character, DBCS, or hexadecimal fields, or for *CALC fields that result in one of these types. If no value is specified, the CCSID is determined based on the CCSIDs of the fields and/or literal values specified on the MAPFLD definition.

Literal values in the MAPFLD definition are tagged with the job default CCSID. However, if the MAPFLD definition consists of only a literal value and the user specifies a field CCSID value, the literal will be tagged with that CCSID. This allows the user to tag a literal with a CCSID other than the job default CCSID.

Note:

Normally, *HEX and *VHEX fields do not have an associated CCSID. Because of this, the data in the field is treated the same regardless of the default CCSID of the system that the data is being used on. However, if the user specifies a CCSID for a *HEX or *VHEX field, the CCSID overrides the hexadecimal attribute of the field (causing the field to be treated as *CHAR/*VCHAR), and the data in the field may be treated differently if it is moved to a system that has a different default CCSID.

For more information on CCSIDs, see the Globalization topic in the Information Center.

An example of the MAPFLD parameter, showing the use of field-type, field-length, field-decimals, and field-CCSID, is as follows:

```
MAPFLD((UNTPRICE 'TOTPRICE / UNTCOUNT' *DEC 7 2)
(SHORTSTR '%SST(LONGSTR 1 5)' *CHAR 5 *N 930))
```

IGNDECERR

Specifies whether the system ignores decimal data errors during query file processing.

***NO:** The system does not ignore decimal data errors.

***YES:** The system ignores decimal data errors. When errors in decimal data are encountered, the invalid sign and/or digits are automatically changed to valid values.

OPNID

Specifies the identifier used to name the query file so it is referred to by the Close File (CLOF) command when it is closed or by the Position Database File (POSDBF) command when it is opened. The identifier must differ from the OPNID associated with any other file which was previously opened by using the OPNDBF command or OPNQRYF command, and which is not yet closed.

***FILE:** The name of the first or only file specified on the FILE parameter is used for the open query file identifier.

open-identifier-name: Specify the name to associate with this open query file.

SEQONLY

Specifies whether sequential-only processing is used for the query file, and also specifies the number of records that can be processed as a group when read or write operations are performed on the open query file. The open query file open data path (ODP) uses a different SEQONLY value than the one specified on this parameter. It depends on other parameter values specified on the OPNQRYF command. A message is sent if the SEQONLY value is changed. More information about sequential-only processing is in the SEQONLY parameter description on the OVRDBF (Override Database File) command, and in the Database Programming topic in the Information Center.

Element 1: Sequential-Only Processing is Used

***YES:** The open query file uses sequential-only processing. Number of records can be specified with this value.

Element 2: Number of Records That can be Processed

number-of-records: Specify the number of records that are processed as a group when read or write operations are performed with the open query file. If no number-of-records value is specified, the system calculates the number of records processed as a group.

Other Single Values

***NO:** The file does not use sequential-only processing.

COMMIT

Specifies whether the SQL statements are run under commitment control.

Before a database file is opened under commitment control, the user must ensure that all files in the commitment transaction are journaled. If only the after images are being journaled, the system implicitly begins journaling both the before and the after images for the duration of the changes being made to files opened under this commitment definition.

***NO:** The open query file is not placed under commitment control.

***YES:** The open query file is placed under commitment control.

OPNSCOPE

Specifies the extent of influence (scope) of the open operation.

***ACTGRPDFN:** The scope of the open operation is determined by the activation group of the program that called the OPNQRYF command processing program. If the activation group is the default activation group, the scope is the call level of the system program performing the open operation. If the activation group is a non-default activation group, the scope is that activation group. In a multi-threaded job, only those opens within the same thread and within the same activation group can share this ODP.

***ACTGRP:** The scope of the open data path (ODP) is the activation group. Only those shared opens from the same activation group can share this ODP. In a multi-threaded job only those

shared opens from the same activation group within the same thread can share this ODP. This ODP is not reclaimed until the activation group is deactivated, or until the Close File (CLOF) command closes the file.

***JOB:** The scope of the open data path (ODP) is the job in which the open operation occurs. If the job is multi-threaded, only those opens from the same thread can share this ODP.

DUPKEYCHK

Specifies whether duplicate key feedback is returned on input/output (I/O) operations. Duplicate key feedback should be requested only for files that are processed by COBOL programs since this is the only high-level language (HLL) that utilizes the feedback. Duplicate key feedback should only be requested when it is used by the COBOL application since providing it can cause a performance degradation. More information on duplicate key feedback is in the Database Programming topic in the Information Center.

***NO:** Duplicate key feedback is not returned on I/O operations.

***YES:** Duplicate key feedback is returned on I/O operations.

ALWCPYDTA

Specifies whether the system is allowed to copy data from the files, file members, and record formats specified on the FILE parameter. If so, the system is allowed to open the query file to the copy. The system tries to avoid using a copy of the data because a copy does not reflect changes made to the database after the information is copied. However, certain requests (such as when key fields contained in multiple based-on files for a join are specified) require that the data be copied before the specified query functions are performed.

***YES:** The system may use a copy of data from the files, file members, and record formats specified on the FILE parameter. A copy of the data is used only when it is needed to perform the requested query functions.

***NO:** The system does not use a copy of data from the files, file members, and record formats specified on the FILE parameter. If it is necessary to use a copy of the data to perform the requested query functions, the query file is not opened and an error message is sent.

***OPTIMIZE:** The system uses a sort routine to order the output from the files, file members, and record formats specified on the FILE parameter. A sort routine is used only if the KEYFLD parameter is specified, and if using a sort routine would improve query performance without conflicting with other OPNQRYF options.

A sort will improve the performance of a query that returns most or all of the records in the file or files specified on the FILE parameter.

Using a sort can increase the time required for the OPNQRYF command to process. This occurs because the sort is performed and all records to be returned through the query are processed while the OPNQRYF command is active. However, because the records are already processed, the reading of the records (by using either a program or the CPYFRMQRYP command) is very fast. Therefore, the overall time to process the query is reduced.

Specifying the KEYFLD parameter for the OPNQRYF command does not ensure that the query will use an index if ALWCPYDTA(*OPTIMIZE) is specified. If a sort routine is used, the file is not opened with indexed access. If the program reading the records from the OPNQRYF command requires indexed access (random processing rather than sequential processing), ALWCPYDTA(*YES) or ALWCPYDTA(*NO) should be specified.

When a sort is used, the query file's position is not changed when a ROLLBACK command is issued. Therefore, when a query is opened that has parameters, ROLLBACK commands that follow do not reset the queried file's position to where it was at the start of the unit of recovery.

Note:

Do not specify ALWCPYDTA(*OPTIMIZE) if you require that a ROLLBACK command reposition the query file, or if you require that the queried file be opened with indexed access.

The following items are required before a sort is valid for the OPNQRYF command:

- ALWCPYDTA(*OPTIMIZE) must be specified.
- The OPTION parameter, if specified, must be *INP.
- A value other than *FILE or *NONE must be specified on the KEYFLD parameter.
- The UNIQUEKEY parameter must not be specified, or must specify *NONE.
- The SEQONLY parameter, if specified, must be *YES.
- If COMMIT(*YES) is specified, the level of record locking (LCKLVL parameter on the STRCMTCTL command) must not be *ALL.
- The DUPKEYCHK parameter must not be specified, or must specify *NO.
- The total buffer length of all fields in the file specified on the FORMAT parameter (or FILE parameter, if the FORMAT parameter is not specified) must not exceed 32700 bytes.

The query optimizer determines whether a sort is used. This decision is based on the number of records expected from the query and the options specified on the OPNQRYF statement. The following items influence the optimizer's choice of a sort:

- The OPTIMIZE parameter should specify *ALLIO or *MINWAIT. If *FIRSTIO is specified, the number of records specified should be close to or equal to the number of result records expected from the query.
- The number of records in a file specified on the FILE parameter should contain a minimum of 200 records.
- The query result should contain a minimum of 200 records.

OPTIMIZE

Specifies the most efficient way the system can perform the selection and join processing necessary to satisfy the other parameter specifications on the OPNQRYF command.

If the KEYFLD or GRPFLD parameters require that an access path be built (when no existing access path is shared), the access path is built completely, regardless of the OPTIMIZE entry. Optimization primarily affects system operation when selection processing is performed.

***ALLIO:** The system attempts to reduce the total input/output time required to process the whole query, assuming that all query records are read from the file.

Element 1: Time Reduction Options

***FIRSTIO:** The system attempts to reduce the input/output time required to open the query file and to retrieve the first buffer of records from the file.

***MINWAIT:** The system attempts to reduce the time required to open the query file by minimizing delays when records are read from the file.

Element 2: Number of Records to Retrieve

number-of-records: Specify the number of query records to use in the optimize operation. This value is ignored when the value *MINWAIT is specified.

OPTALLAP

Specifies whether the query optimizer should consider all the access paths that exist over the files being queried when determining how to implement the query.

***NO:** Allow the query optimizer to operate normally. When determining how to implement a query, the optimizer considers access paths until an internal timeout value has been exceeded. If there are a large number of access paths over the files being queried, the optimizer may time out before it has considered all the available access paths.

***YES:** Force the query optimizer to ignore the internal timeout value and consider all the available access paths over all the files in the query.

Note: If there are a large number of access paths over the files it may take a long time to optimize the query.

SRTSEQ

Specifies the sort sequence to be used for sorting and grouping selections specified on the QRYSLT or GRPSLT parameters, joins specified on the JFLD parameter, ordering specified on the KEYFLD parameter, grouping specified on the GRPFLD parameter, %MIN or %MAX built in functions, or unique key values specified on the UNIQUEKEY parameter.

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

***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 query.

LANGID

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

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

language-ID: Specify the language identifier to be used by the job.

TYPE Specifies the recursion level at which the Reclaim Resources (RCLRSC) command closes the file.

Note: This parameter is ignored unless the default value is specified on the OPNSCOPE parameter and the request is from the default activation group.

***NORMAL:** The RCLRSC command closes the file if the program call that ran the OPNQRYF command is ended without closing the file.

***PERM:** The file remains open until the Close File (CLOF) command closes it, or until the routing step or default activation group ends. The query file remains open even if the RCLRSC command is run.

Table 1. Query Field Structure

Type	Field Type	External Field Length	Default Length and Decimals	Internal Field Length in Bytes
Binary				
Binary				
Floating-point				
Floating-point				
Packed decimal				
Zoned decimal	*BIN2	1-5	5 0 ²	2
Character	*BIN4	1-10	10 0 ²	4
Variable length character	*FLT4	1-9	7 6 ¹	4
Hex	*FLT8	1-17	15 14 ¹	8
Variable length hex	*DEC	1-31	15 5 ¹	1-16
DBCS-only	*ZONED	1-31	15 5 ¹	1-31
DBCS-either	*CHAR	1-32766	32	1-32766
DBCS-open	*VCHAR	0-32740	32	0-32740
DBCS-graphic	*HEX	1-32766	32	1-32766
Variable length	*VHEX	0-32740	32	0-32740
DBCS-only	*ONLY	4-32766 ³	32	4-32766
DBCS-either	*EITHER	4-32766 ³	32	4-32766
Variable length	*OPEN	4-32766	32	4-32766
DBCS-open	*GRAPHIC	1-16383 ⁵	32	2-32766
DBCS-graphic	*VONLY	0-32740	32	0-32740
Variable length	*VEITHER	0-32740	32	0-32740
DBCS-only	*VOPEN	0-32740	32	0-32740
DBCS-either	*VGRAPHIC	0-16370 ⁵	32	0-32740
Date	*DATE	5-10 ⁴	8	4
Time	*TIME	4-8 ⁴	7	3
Timestamp	*TIMESTP	14; 16-26 ⁴	26	10

- ¹ If the number of decimal digits is specified, but the decimal precision value is omitted for an *FLT4, *FLT8, *DEC, or *ZONED field, the precision defaults to zero decimal digits.
- ² A nonzero value is not allowed for the decimal precision of a binary number.
- ³ The field length must be an even-numbered value.
- ⁴ These fields can be longer if they are padded on the right with blanks.
- ⁵ These field lengths are in a number of graphic characters.

Examples for OPNQRYF

Example 1: Selecting Specific Records

Note: Additional examples of selecting records using the OPNQRYF command can be found in the Database Programming topic in the Information Center.

```
OPNQRYF FILE(ordfile) OPTION(*all)
  QRYSLT('orddate=%range("840101" "841231")
  & ordamt>100')
  KEYFLD((ordamt *descend))
```

This command uses the QRYSLT parameter to select only records in the first member of file ORDFILE that have an order date in 1984 and an order amount greater than 100. Because the FORMAT parameter

is omitted, the open query file has the same record format as file ORDFILE. The open query file allows all file operations (input, output, update, and delete). The KEYFLD specification is used to force the records to be arranged by descending value of order amount.

Example 2: Using the %XLATE Built-In Function

```
OPNQRYF FILE(telefile)
  QRYSLT('%xlate(username qsysrntb1) *ct "GEORGE"')
```

This command uses the %XLATE built-in function to translate the field USRNAME to uppercase, and to instruct the *CT operator to select only records that contain the value GEORGE in the field USRNAME. QSYSRNTBL is an IBM-supplied system translation table that converts lowercase alphabetic (a through z) to uppercase (A through Z). The translation is done to ensure that the search value is recognized even if its characters appear in mixed case. The records available through the open query file have the same record format as those in file TELEFILE.

Example 3: Using the %XLATE Built-In Function

```
OPNQRYF FILE(telefile)
  QRYSLT('username *ct ''GEORGE''')
  MAPFLD((username
  '%xlate(telefile/username qsysrntb1)'))
```

In the previous example, the value of field USRNAME, which is returned to the high-level language (HLL) program that reads records from the open query file, is not translated to uppercase.

This example shows a way to make the uppercase version of field USRNAME available to the HLL program. This is done by defining a mapped field (MAPFLD parameter) for the translated value of field USRNAME. The field has the same field name as the field name in the open query file record format being used. The translated version of the field is used for selection (QRYSLT parameter) and is used in the open query file record format.

Example 4: Using the %SST Built-In Function

```
OPNQRYF FILE((histlib/ordfile hist1))
  OPTION(*inp *upd *dlt)
  FORMAT(ordinfo orddtls)
  QRYSLT('month=7')
  MAPFLD((year '%sst(orddate 1 2)' *zoned 2)
  (month '%sst(orddate 3 2)' *zoned 2)
  (day '%sst(orddate 5 2)' *zoned 2))
```

This command uses the %SST built-in function to create a substring of the year, month, and day parts of character field ORDDATE in file ORDFILE. The command also maps a character string to a zoned field, which is valid as long as the zoned field has the same length as the character string. If the file ORDINFO has a record format, ORDDTLS, containing at least the field's YEAR, MONTH, and DAY records, these fields have input-only usage in the open query file record format because they are defined by using a built-in function (%SST) and are mappings that mix character and numeric (zoned decimal format) types. The file is opened for input, update, and delete operations, but none of the field's YEAR, MONTH, and DAY records are updated using the open query file open data path (ODP). The open query file uses only records in the HIST1 member of file ORDFILE in library HISTLIB, and the records retrieved through the file have the same format as record format ORDDTLS in file ORDINFO. Only records pertaining to the month of July are processed through the open query file (QRYSLT parameter).

Example 5: Returning the First Record of Each Set

```
OPNQRYF FILE((routelf *first locusr))
  QRYSLT('%sst(toloc 1 4) *eq "ROCH"')
  KEYFLD(fromusr fromloc tousr toloc)
  UNIQUEKEY(*all)
```

This command uses the KEYFLD and UNIQUEKEY parameters to return only the first record of each set of records in record format LOCUSR in the first member of file ROUTELF that have the same values for the fields FROMUSR, FROMLOC, TOUSR, and TOLOC. The query result is further restricted by selecting only records that have the value ROCH in the first four characters of field TOLOC. The records available through the open query file contain all of the fields in record format LOCUSR of file ROUTELF. If the file ROUTELF contains information about messages routed by an application, this example identifies all unique sender and receiver pairs in which the receiving location name begins with ROCH.

Example 6: Joining a File to Itself

```
OPNQRYF FILE(partpf partpf)
  FORMAT(partjoin)
  JFLD((1/pnbr 2/pnbr *GE))
  MAPFLD((pnm1 '1/pname') (pnm2 '2/pname')
  (pnbr '1/pnbr'))
```

This example illustrates how a file is joined to itself, as well as how to use the MAPFLD parameter to rename fields in the based-on files. A greater than or join is performed using field PNBR as both the join from-field and the join to-field.

The format of file PARTJOIN is assumed to contain fields named PNBR, PNM1, and PNM2. The field name PNBR is valid in the query output record format because that field is defined on the MAPFLD parameter. If the record format in file PARTJOIN contains a field named PNAME, an error occurs because the field exists in both files specified on the FILE parameter, and is not the name of a field defined on the MAPFLD parameter. The mapped field definitions are field names, so the attributes of fields PNM1 and PNM2 match the attributes of field PNAME, and the attributes of field PNBR in the open query file records match field PNBR in file PARTPF. Further, when a file is joined to itself, it is always necessary to specify a file number name for any field that is defined in the based-on file.

Example 7: Renaming Fields in Based-On Files

The same query can also be specified as follows:

```
OPNQRYF FILE(partpf partpf)
  FORMAT(partjoin)
  QRYSLT('1/pnbr *GE 2/pnbr')
  MAPFLD((pnm1 '1/pname') (pnm2 '2/pname')
  (pnbr '1/pnbr'))
```

Because more than one file is specified on the FILE parameter, and the default value is specified for the JDFTVAL and JORDER parameters, the system takes the join specifications from the values specified on the QRYSLT parameter.

Example 8: Selecting Master Records With No Detail Records

```
OPNQRYF FILE(cusmas ordfil)
  FORMAT(cusmas)
  JFLD((cusnbr ordfil/cusnbr))
  JDFTVAL(*onlydft)
  MAPFLD((cusnbr 'cusmas/cusnbr'))
```

This command uses a join query to select only master records that have no associated detail records. The master file (CUSMAS) is joined (equal join) to the detail file (ORDFIL) by the customer number field that appears in both record formats. The customer number field name is the same in both record formats (CUSNBR). Because CUSNBR is the name of a field defined on the MAPFLD parameter, everywhere the simple field name CUSNBR is used, the mapped field version of the CUSNBR field in file CUSMAS is used (including the open query file record format, which matches the customer master file record format). The JDFTVAL parameter indicates that only records that are produced by using default values are

available through the open query file. Every master record that has associated detail records (with the same value of the customer number field) is excluded, and every master record that has no associated detail records creates a result record.

Example 9: Identifying Detail Records With No Associated Master Record

```
OPNQRYF FILE(ordfil cusmas)
  FORMAT(ordfil)
  JFLD((cusnbr cusmas/cusnbr))
  JDFTVAL(*onlydft)
  MAPFLD((cusnbr 'ordfil/cusnbr'))
```

This change of the previous example (using the same files) shows how to identify all detail records with no associated master record (in this case, all orders with an unregistered customer number):

Example 10: Calculating Basic Statistics

```
OPNQRYF FILE(scores)
  FORMAT(clsstats)
  GRPFLD(clsid)
  GRPSLT('clsavg<70 & clsmax-clsmin>30')
  MAPFLD((clsCnt '%count')
    (clsavg '%avg(usrscore)')
    (clsmin '%min(usrscore)')
    (clsmax '%max(usrscore)'))
```

This command uses the grouping function to calculate basic statistics for each group of records in file SCORES that have the same value in the field CLSID. Assuming file CLSSTATS has a record format containing field CLSID and all fields specified on the MAPFLD parameter, each record available through the open query file contains the value of the grouping field (CLSID) as well as the number of records included in the group and the average, minimum, and maximum values of field USRSCORE in the group. Selection occurs after grouping, so that records are created for groups only when the average value of USRSCORE in the group is less than 70 and the difference between the maximum and minimum scores in the group is greater than 30.

Example 11: Selecting Records With a Specific Value

```
OPNQRYF FILE(itmmast)
  QRYSLT('itmcode=%range(32 50) & itmtype="P"')
  ALWCPYDTA(*no)
  OPTIMIZE(*firstio)
  SEQONLY(*yes 10)
  TYPE(*perm)
```

This command selects from the first member of file ITMMAST only the records that have a value of field ITMCODE in the range from 32 through 50 and also have a value of field ITMTYPE equal to the letter P. The ALWCPYDTA parameter specifies that the open query file must never use a copy of the records in file ITMMAST. The OPTIMIZE and SEQONLY parameter values cause the system to attempt to improve processing for the open query file to minimize the time needed to retrieve the first buffer of ten records. This combination of parameter values is a good choice if the file is used with a high-level language interactive inquiry program that shares the open query file open data path (ODP) and shows ten records on each display screen. The open data path (ODP) for the open query file is 'permanent' (TYPE parameter), which means that it remains open either until the file is closed by using the Close File (CLOF) command or until the routing step ends.

Example 12: Tagging a Literal with a Specific CCSID

```
OPNQRYF FILE(itmmast)
  QRYSLT('itmtype=pfield')
  MAPFLD((pfield 'P' *CHAR 1 *N 930))
```

This command selects from the first member of file ITMMAST only the records that have a value of field ITMTYPE equal to the letter 'P' in character set 930. The mapped field is created so that the literal 'P' can be tagged with a specific CCSID.

If a literal is not tagged with a specific CCSID, it is assigned the CCSID of the job running the query. Because of this, if an OPNQRYF statement is part of a CL program that is shared among systems with differing CCSIDs (in different countries, perhaps), a query that uses a literal in the selection specifications may not return the same results on all systems, even though the data in the files is the same. This happens because the internal representation of the literal may be different when the CL program is run in a job with a different CCSID. This representation then may not match the same records in the file. Note that the internal representation of the data in the file does not change based on the CCSID of the current job.

Tagging the literal with a specific CCSID avoids this problem. A literal tagged with a specific CCSID keeps the same internal representation on all systems. The CCSID that is used to tag the literal should be the same as the CCSID assigned to the field against which the literal is being compared.

Example 13: Using a Nonjoin Query

```
OPNQRYF FILE((EMPLOYEE)) KEYFLD((NAME))
        ALWCPYDTA(*OPTIMIZE)
```

This command returns all of the records in the EMPLOYEE file.

Example 14: Using a Join Query

```
OPNQRYF FILE((EMPLOYEE) (MANAGEMENT))
        FORMAT(EMPLOYEE) KEYFLD((NAME))
        JFLD((1/EMPID 2/MEMPID)) ALWCPYDTA(*OPTIMIZE)
```

This command returns all of the records required by the join criteria.

Additional Considerations

The file, library, and file member names used by the open data path (ODP) are the same as the first file and file member names specified on the FILE parameter, unless an override forces the use of a different file or file member name. The record format name of the open query file is the same as that specified on the FORMAT parameter.

The OPNQRYF command always opens a file with an open data path (ODP) that is shared, as if SHARE(*YES) is specified for the file. If the file, library, or file member name specified in the HLL program differs from the name of the open query file, an override command must be used to specify the correct file, library, and member names to allow the high-level language program to share the open query file ODP. If the first, or the only, member queried has an attribute of SHARE(*NO), SHARE(*YES) must be specified in an override to enable an HLL program to share the query file ODP.

If the OPNQRYF is scoped to the job, any subsequent open, other than a query open, of the same file can share the ODP whether scoped to an activation group or the job. If the OPNQRYF is scoped to an activation group, any subsequent open, other than a query open, of the same file can share the ODP if it is also scoped to the same activation group.

The open query file ODP also has a LVLCHK attribute that matches the first, or only, member used for the query. Shared opens of an open query file are level checked unless the first queried member has an attribute of LVLCHK(*NO) or LVLCHK(*NO) is specified either in the program that opens the file or in an override which occurs before the shared open. The level number for the open query file record format is the same as the record format identified on the FORMAT parameter.

An expanded discussion of field names, expressions, and built-in functions, and restricted built-in functions used with parameters on the OPNQRYF command follows:

Field Names

The field name used as the first part of an element in the list specified on the MAPFLD parameter must be a simple name, and the field names in the record format identified on the FORMAT parameter are always treated as simple names. Any other field name specified on an OPNQRYF command parameter (QRYSLT, KEYFLD, JFLD, GRPFLD, GRPSLT, or the field-definition expression part of the MAPFLD parameter) is a qualified field name, specified as follows:

field-name

Specify a simple field name that identifies a field that is defined on the MAPFLD parameter, or with a field name that is unique among all field names from all record formats included in the list specified on the FILE parameter. This form is not allowed if there is no MAPFLD parameter definition for the specified field name and the FILE parameter includes more than one record format that contains a field with the specified name, even if the same file and record format is specified more than once in the list on the FILE parameter.

For example, AMOUNT is valid if the field named AMOUNT is defined on the MAPFLD parameter. It is also valid if AMOUNT is not defined on the MAPFLD parameter, as long as there is only one field named AMOUNT in any record format specified on the FILE parameter.

file-name/field-name

Specify a field name that is qualified with the simple name of the file specified on the FILE parameter whose record format contains the field, but only if the simple file name is unique among all file names specified on the FILE parameter. This form is not allowed if the same simple file name is specified more than once in the list specified for the FILE parameter, even if different library, member, or record format names are used.

For example, WHS01/PARTNBR is valid if there is a field named PARTNBR in the record format for file WHS01, and file name WHS01 is only specified once on the FILE parameter.

file-nbr/field-name

Specify a simple field name that is qualified with the number of the element in the FILE parameter list for the record format that contains the field. The file-nbr qualifier must be specified without leading zeros. This form is only required if the same simple file name is specified more than once in the list specified on the FILE parameter.

For example, 2/BALDUE is valid if the second file record format in the list specified on the FILE parameter contains a field named BALDUE.

*MAPFLD/field-name

Specify a simple field name that is qualified with the special value *MAPFLD if the field is defined on the MAPFLD parameter. When the field is defined, this form has the same meaning as specifying the simple field name with no qualifier. If the field is not defined on the MAPFLD parameter, *MAPFLD cannot be specified.

For example, *MAPFLD/AVGBAL is valid if the AVGBAL field is specified as the first part of one of the mapped field list elements specified on the MAPFLD parameter.

Expressions

Expressions specified on the QRYSLT, GRPSLT, and MAPFLD parameters are similar to expressions specified on other CL command parameters. Logical, relational, numeric, and string operations are performed by using combinations of field values and constants. Symbolic and named operators are supported, as well as many built-in functions, and parentheses are used to control the order of evaluation.

There are also differences in the expressions specified on OPNQRYF parameters and on other CL command parameters. The following list summarizes the ways that expressions on the QRYSLT, GRPSLT, and MAPFLD parameters differ from normal CL expressions:

- The expression string must be enclosed in apostrophes if it contains embedded blanks or special characters
- The following differences affect numeric and string literals:
 - Character string constants are quoted by using apostrophes or double quotes
 - The leading and trailing zeros of a numeric constant are significant parts of its attributes
 - Floating-point constants (including the special values *INF and *NEGINF) are used in expressions
- The following differences contrast CL variables with database fields:
 - No prefixed ampersand (&) is used in database field names
 - Qualified field names are supported
 - No 'logical' field type exists for database fields
 - Many additional data types are supported for database fields
- The following CL operators are not supported on the OPNQRYF command:
 - *BCAT or v>
 - *TCAT or v<
- The following additional operators are supported beyond CL support:
 - // for remainder
 - ** for exponentiation
 - *CT for 'contains' (character scan)
 - *XOR or && for 'logical exclusive or'
- The following differences affect built-in function support:
 - The %SWITCH built-in function is not supported
 - Many additional built-in functions are supported
 - Nested built-in functions and expressions for built-in function arguments (such as '%LOG(%SIN(x))') generally are allowed
 - To support expressions as built-in function arguments, any argument that is a signed numeric value or an expression (for example, '%MIN(3 (-2) x (y+4))') must be enclosed in parentheses

The following table shows the priority of all operators that are used for expressions on the QRYSLT, GRPSLT, or MAPFLD parameters. Only operators listed for priorities 1 through 5, excluding the *NOT and ~ operators, are allowed in an expression specified on the MAPFLD parameter:

Priority	Operators
1	+, - (when used for signed numeric values), *NOT, ~
2	**
3	*, /, // (a / must have a space before the / and/or after the /)
4	+, - (when used between two operands)
5	*CAT, v v
6	*GT, *LT, *EQ, *GE, *LE, *NE, *NG, *NL, *CT, >, <, =, >=, <=, ~>, ~<
7	*AND, &
8	*OR, *XOR, v, &&

Except for operators ~ and *NOT, the operators for priorities 1 through 4 are numeric operators, which require numeric operands. The operators for priority 5 are string operators, which require operands to be either character or DBCS strings. Priority 6 operators are called relational operators, which require at least one operand that is a field name or a numeric or string expression (not a constant). The operators for

priorities 7 and 8, plus the \neg and *NOT operators (priority 1), are logical operators. The operands in a logical expression are relations (constructed by using a relational operator with appropriate operands) and other logical expressions.

The operands in a string expression, including string operands for a built-in function, are a combination of character fields and DBCS fields and constants. If both operands of such an expression are DBCS-only fields or constants, the final result from evaluation of the expression is a DBCS-only field value. If the operands are a combination of DBCS or character fields or constants, the result is a DBCS-open field value. When DBCS fields are concatenated, the extraneous shift-in and shift-out characters between the fields are removed.

The result produced by a + or - sign prefixed operator has the same attributes as the operand, unless the operand of a - sign prefixed operator is a *BIN2, in which case the result is a *BIN4. The result of an ** operator (exponentiation) is a double-precision floating-point number (*FLT8). For other numeric operators that require two operands, if either operand is a floating-point number, the result is a double-precision floating point number (*FLT8). If both operands are fixed-point numbers, the system uses the information in the following table to determine the number of total and fractional digits required to produce a packed decimal (*DEC) result. If both operands are zero-precision binary fields and/or integer constants, the result is a *BIN4, unless the operator is a "/". In that case, the result is the same as for a fixed-point result. If the total number of digits required exceeds 31, the number of fraction digits is reduced enough to enable calculation of the result with a total of 31 digits. If some fraction digits are dropped and the attributes of the end result of the computation (the attributes specified on the MAPFLD parameter for the field) require greater precision than that of the intermediate result, a warning message is sent to indicate that some precision was lost in evaluating the expression.

Operation	Result (Total Digits)	Result (Fractional Digits)
+	$\text{MAX}(d1-f1, d2-f2) + \text{MAX}(f1, f2) + 1$	$\text{MAX}(f1, f2)$
-	$\text{MAX}(d1-f1, d2-f2) + \text{MAX}(f1, f2) + 1$	$\text{MAX}(f1, f2)$
*	$d1 + d2$	$f1 + f2$
/	31	$31 - (d1 - f1 + f2)$
//	$\text{MIN}(d1-f1, d2-f2) + \text{MAX}(f1, f2)$	$\text{MAX}(f1, f2)$

Legend:

- d1** Total digits in operand 1
- f1** Fractional digits in operand 1
- d2** Total digits in operand 2
- f2** Fractional digits in operand 2

When a numeric or string expression is specified on the MAPFLD parameter, the attributes of the final result are used in one of two ways. They are either used directly for the field value (if field-type *CALC is specified and the field is not contained in the prototype record format identified on the FORMAT parameter), or the final result is changed to match the attributes specified on the MAPFLD parameter or contained in the field definition in the record format identified by the FORMAT parameter.

Both operands of a relational operator can be constants. The fields, constants, or expressions specified as operands on the left and right side of a relational operator must be of the same type, either numeric or string. Any combination of character and DBCS field operands are allowed except that a character field cannot be related to a DBCS-only field.

There are two types of DBCS constants: DBCS-only and DBCS-open. A DBCS-only constant has only DBCS data between its apostrophes. This data must be enclosed in SO/SI characters. A DBCS-open

constant has a mixture of DBCS and alphanumeric data. An SO character (HEX 0E) indicates the start of a group of DBCS characters and an SI character (HEX 0F) follows the last double-byte character of the group.

If a numeric or string expression appears as a complex selection operand on the QRYSLT or GRPSLT parameters, attributes of the final result of the expression used for the selection operand are changed to match the other relational operand.

It is not necessary for operands of a relational operator to have identical attributes, but numeric operands cannot be mixed with character operands. If the operands do not have identical attributes, the system changes them to identical attributes (except for the *CT operator, where the character string operands may be of different lengths), before performing the operation. This change uses packed decimal format if both operands are fixed-point numeric operands, or floating-point format if either operand is a floating-point number. The changes for fixed-point numeric operands align their decimal points and pad them with zeros. Numeric type changes may truncate fractional digits if more than 31 total digits are required for fixed-point numbers, or may drop some of the least significant digits if more than 15 total digits are required for floating-point numbers. Character operands are changed by padding the shorter operand with blanks.

The *CT operator performs a scan of the character field or string expression (except for expressions made up of a single character string literal) that must be specified as the left side of the relation, in order to determine if it contains the character string, field, or expression value specified as the right side of the relation. The second operand (the search value) must be no longer than the first operand (the base string).

If the string is found, the relation is satisfied and the result is a logical value of 'true'; otherwise, the result is a logical 'false' value. The following example illustrates this process:

Field BASEFLD contains the value 'THIS IS A TEST'.

Field TESTFLD contains the value 'TE'.

Expression	Result
'BASEFLD *CT "IS A"'	True
'BASEFLD *CT TESTFLD'	True
'BASEFLD *CT "X"'	False
'BASEFLD *CT TESTFLD v v "Z"'	False
'BASEFLD v v "ABC" *CT "TAB"'	True

Built-in Functions

The built-in functions listed below are supported for the expression used to define a derived field on the MAPFLD parameter or a complex selection operand specified on the QRYSLT or GRPSLT parameters.

A numeric argument is a numeric field, a numeric constant or a numeric expression. A string argument is a character field, a character string literal, or a string expression. Unless otherwise noted, all built-in functions allow expressions, including other built-in functions, to be used as arguments.

For a field that appears in the record format identified by the FORMAT parameter, and that is also defined by an expression on the MAPFLD parameter, the expression result is calculated by using the attributes described below. Then the resultant value is mapped to match the attributes of the field.

%ABSVAL (numeric-argument)

%ABSVAL accepts a numeric argument and returns the absolute value of the argument. The returned value has the same attributes as the argument, unless the argument is a *BIN2, in which case the returned value is a *BIN4.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a packed decimal number (*DEC) with 8 digits and 0 precision (date duration), 6 digits and 0 precision (time duration), or 20 digits and 6 precision (timestamp duration).

%ACOS (numeric-argument)

%ACOS accepts a numeric argument and returns the arc cosine of the argument, in radians. %ACOS and %COS are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%AND (string-argument ...)

%AND accepts two or more character or hexadecimal string arguments and returns a string that is the bit-wise 'AND' (logical and) of the arguments. This function takes the first argument string, ANDs it with the next string, and continues to AND each successive argument with the previous result. If an argument is encountered that is shorter than the previous result, it is padded on the right with blanks. The returned value is a string of type *HEX with the same length as the longest argument. If any of the arguments are variable-length, the maximum length is used as the length of the argument.

%ANTILOG (numeric-argument)

%ANTILOG accepts a numeric argument and returns the antilogarithm (base 10) of the argument. %ANTILOG and %LOG are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%ASIN (numeric-argument)

%ASIN accepts a numeric argument and returns the arc sine of the argument, in radians. %ASIN and %SIN are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%ATAN (numeric-argument)

%ATAN accepts a numeric argument and returns the arc tangent of the argument, in radians. %ATAN and %TAN are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%ATANH (numeric-argument)

%ATANH accepts a numeric argument and returns the hyperbolic arc tangent of the argument, in radians. %ATANH and %TANH are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%AVG (numeric-argument)

%AVG accepts a numeric argument and returns the average value of its argument for the group of records defined on the GRPFLD parameter. The argument must be a field name or an expression (not a literal).

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. If no records are selected, the result is the null value. Otherwise,

- If the argument is fixed-point, the result is a packed decimal number (*DEC) with 31 total digits and the same number of integer digits as the argument.
- If the argument is floating-point, the result is a double-precision floating-point number (*FLT8).
- If the argument is date duration, time duration, or timestamp duration, the returned value is a packed decimal number (*DEC) with 31 digits and 0 precision (date duration), 31 digits and 0 precision (time duration), or 31 digits and 6 precision (timestamp duration).

%AVG is an aggregate function that is used for a nongrouping field in a query that uses the grouping function.

%CHAR (date/time-argument date/time-format)

%CHAR accepts a date/time argument and date/time format and returns the character representation of the argument using the specified format. The date/time argument can be a date, time, or timestamp field. The returned value is of type *CHAR and is tagged with the CCSID of the current job.

The date/time format is optional. If specified, it must be one of the following:

- EUR** European format
- ISO** International Standards Organization format
- JIS** Japanese Industrial Standard format
- USA** United States format

If the format is not specified, the job default format is used.

```
Example:
OPNQRYF
  FILE(library/file)
  GRPFLD(charfld)
  GRPSLT('charfld = %CHAR(timefld "USA")')
```

%COS (numeric-argument)

%COS accepts a numeric argument and returns the cosine of the argument. The argument must be specified in radians. %COS and %ACOS are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%COSH (numeric-argument)

%COSH accepts a numeric argument and returns the hyperbolic cosine of the argument. The argument must be specified in radians.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%COT (numeric-argument)

%COT accepts a numeric argument and returns the cotangent of the argument. The argument must be specified in radians.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%COUNT

%COUNT does not support any arguments. It returns the count of the number of records contained in the group of records defined on the GRPFLD parameter. The returned value is a

4-byte binary number (*BIN4) with 10 total decimal digits and no fraction digits. %COUNT is an aggregate function that applies only to a query that uses the grouping function.

%CURDATE

%CURDATE does not support any arguments. It obtains the current date based on a reading of the time-of-day clock. The returned value is of type *DATE. The format and separator are derived from the job attributes.

%CURSERVER

%CURSERVER does not support any arguments. If only non-distributed files are specified then it obtains the current server name (or RDB name) of the local system. If distributed files are specified then it obtains the current server name (or RDB name) of the COORDINATOR node. The returned value is of type variable-length character (*VCHAR) with a maximum length of 18.

%CURTIME

%CURTIME does not support any arguments. It obtains the current time based on a reading of the time-of-day clock. The returned value is of type *TIME. The format and separator are derived from the job attributes.

%CURTIMESTP

%CURTIMESTP does not support any arguments. It obtains the current timestamp based on a reading of the time-of-day clock. The returned value is of type *TIMESTP. The format and separator will be derived from the job attributes.

%CURTIMEZONE

%CURTIMEZONE does not support any arguments. It obtains the current time zone. The returned value is a packed decimal number (*DEC) with 6 digits and 0 precision.

%DATE (date/time-argument)

%DATE accepts a date/time argument and returns a date. The date/time argument can be a date or timestamp field, a character or hexadecimal field containing the external form of a date, a date literal, or a numeric field or literal value in the range 1 - 3,652,059. The returned value is of type *DATE.

Example:

```
OPNQRYF
  FILE(library/file)
  QRYSLT ((' %DATE(tstampfld) =
    "1989-10-23" ))
```

%DAY (date/time-argument)

%DAY accepts a date/time argument and returns the day part of the value. The date/time argument can be a date or timestamp field, a date duration or timestamp duration (field or literal), or a numeric field or literal. The returned value is of type *BIN4.

A numeric field argument must be defined as packed decimal (*DEC) with 8 digits and 0 precision for date duration or packed decimal (*DEC) with 20 digits and 6 precision for timestamp duration. A numeric constant argument must have 8 digits followed by a decimal point, or 14 digits followed by a decimal point and 6 digits.

%DAYS (date/time-argument)

%DAYS accepts a date/time argument and returns an integer representation of the date. The date/time argument can be a date or timestamp field, a character or hexadecimal field containing the external form of a date, or a date literal. The returned value is of type *BIN4.

%DIGITS (numeric-argument)

%DIGITS accepts a numeric argument and returns a character representation of its numeric value, not including the sign or a decimal point. The result is tagged with the CCSID of the current job. For example, %DIGITS (-1.5) returns the character string 15. The numeric argument must not be a floating point number.

%DURDAY (integer-argument)

%DURDAY accepts an integer argument and returns a labeled duration of days. The integer argument for this function can be a numeric expression, a field, or a literal.

This built-in function is allowed to stand by itself in the *mapped-field-definition* of the MAPFLD parameter, and is allowed as part of an arithmetic (addition or subtraction) expression with a date or timestamp field on the QRYSLT, GRPSLT, or MAPFLD parameters.

%DURHOUR (integer-argument)

%DURHOUR accepts an integer argument and returns a labeled duration of hours. The integer argument for this function can be a numeric expression, a field, or a literal.

This built-in function is allowed to stand by itself in the *mapped-field-definition* on the MAPFLD parameter, and is allowed as part of an arithmetic (addition or subtraction) expression with a time or timestamp field on the QRYSLT, GRPSLT, or MAPFLD parameters.

%DURMICSEC (integer-argument)

%DURMICSEC accepts an integer argument and returns a labeled duration of microseconds. The integer argument for this function can be a numeric expression, a field, or a literal.

This built-in function is allowed to stand by itself in the *mapped-field-definition* on the MAPFLD parameter, and is allowed as part of an arithmetic (addition or subtraction) expression with a timestamp field on the QRYSLT, GRPSLT, or MAPFLD parameters.

%DURMINUTE (integer-argument)

%DURMINUTE accepts an integer argument and returns a labeled duration of minutes. The integer argument for this function can be a numeric expression, a field, or a literal.

This built-in function is allowed to stand by itself in the *mapped-field-definition* on the MAPFLD parameter, and is allowed as part of an arithmetic (addition or subtraction) expression with a time or timestamp field on the QRYSLT, GRPSLT, or MAPFLD parameters.

%DURMONTH (integer-argument)

%DURMONTH accepts an integer argument and returns a labeled duration of months. The integer argument for this function can be a numeric expression, a field, or a literal.

This built-in function is allowed to stand by itself in the *mapped-field-definition* on the MAPFLD parameter, and is allowed as part of an arithmetic (addition or subtraction) expression with a date or timestamp field on the QRYSLT, GRPSLT, or MAPFLD parameters.

%DURSEC (integer-argument)

%DURSEC accepts an integer argument and returns a labeled duration of seconds. The integer argument for this function can be a numeric expression, a field, or a literal.

This built-in function is allowed to stand by itself in the *mapped-field-definition* on the MAPFLD parameter, and is allowed as part of an arithmetic (addition or subtraction) expression with a time or timestamp field on the QRYSLT, GRPSLT, or MAPFLD parameters.

%DURYEAR (integer-argument)

%DURYEAR accepts an integer argument and returns a labeled duration of years. The integer argument for this function can be a numeric expression, a field, or a literal.

This built-in function is allowed to stand by itself in the *mapped-field-definition* value on the MAPFLD parameter, and is allowed as part of an arithmetic (addition or subtraction) expression with a date or timestamp field on the QRYSLT, GRPSLT, or MAPFLD parameters.

Example:

```
OPNQRYF
  FILE((library/file))
  QRYSLT('startfld > %CURDATE + oneyear *AND
        endfld < %CURDATE + %DURYEAR(2)')
  MAPFLD((oneyear '%DURYEAR(1)')
```

%EXP (numeric-argument)

%EXP accepts a numeric argument and returns a value that is the base of the natural logarithm (e) raised to a power specified by the argument. %EXP and %LN are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types may be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%HASH (expression-argument)

%HASH accepts a valid expression and returns a 4-byte binary number (*BIN4) with 10 total decimal digits and no fraction digits. The returned value will be the partition number of the record selected.

A valid expression cannot include aggregate functions such as %COUNT, %AVG, %MIN, %MAX, %SUM, and %STDDEV as operands to %HASH.

Use the %HASH function to determine what the partitions would be if the partitioning key was composed of EMPNO and LASTNAME. The following example returns the partition number for every row in EMPLOYEE.

```
Example:
OPNQRYF
  FILE((CORPDATA/EMPLOYEE))
  FORMAT(FNAME)
  MAPFLD((HASH '%HASH((1/EMPNO) (1/LN))')
```

%HEX (hexadecimal-argument)

%HEX accepts an argument and returns the hexadecimal equivalent of the argument's value. The hexadecimal argument can be of any type. The returned value is of type *CHAR, and is tagged with the CCSID of the current job.

%HOUR (date/time-argument)

%HOUR accepts a date/time argument and returns the hour part of the value. The date/time argument can be a time or timestamp field, a time duration or timestamp duration (either field or literal), or a numeric field or literal. The returned value is of type *BIN4.

A numeric field argument must be defined as packed decimal (*DEC) with 6 digits and 0 precision for time duration or packed decimal (*DEC) with 20 digits and 6 precision for timestamp duration. A numeric constant argument must have 6 digits followed by a decimal point, or 14 digits followed by a decimal point and 6 digits.

```
Example:
OPNQRYF
  FILE(library/file)
  QRYSLT(('%HOUR(timefld2) = 12'))
```

%LEN (length-argument)

%LEN accepts one argument and returns the number of bytes used to represent the value unless the value is a graphic field type. If the value is a graphic field type, the number of graphic characters is returned. The length argument can be of any type. The returned value is of type *BIN4.

```
Example:
OPNQRYF
  FILE(library/file)
  QRYSLT('%LEN(varlenfld) <= 30')
```

Argument Type	Result Length in Bytes
Character	1-32766
Hex	1-32766
DBCS-only	4-32766
DBCS-either	4-32766
DBCS-open	4-32766
DBCS-graphic	1-16383
Variable Character	0-32740

Variable Hex	0-32740
Variable DBCS-only	0-32740
Variable DBCS-either	0-32740
Variable DBCS-open	0-32740
Variable DBCS-graphic	0-16370
Date	4
Time	3
Timestamp	10
Binary *BIN4	2
Binary *BIN8	4
Floating point *FLT4	4
Floating point *FLT8	8
Packed decimal (p,s)	INTEGER(p/2)+1, (1-16)
Zoned decimal (p,s)	p (1-31)

p=precision, s=scale	

String notes:

The %LEN function returns the length of the value as it is stored in the data space.

- For fixed-length fields, the length is always the same as the declared size of the field, not the length of the actual data in the field.
- For variable-length fields, the length is the length of the actual data in the field, including trailing blanks.

For example, assume FIXED10 is a *CHAR(10) field, and VAR10 is a *VCHAR(10) field. The following example shows results of the %LEN function:

%LEN Statement	Field Data	Result
-----	-----	-----
%LEN(fixed10)	'1234567890'	10
%LEN(fixed10)	'12345'	10
%LEN(var10)	'1234567890'	10
%LEN(var10)	'12345'	5
%LEN(var10)	'12345 '	7
%LEN(var10)	' '	0

%LN (numeric-argument)

%LN accepts a numeric argument and returns the natural logarithm of the argument. %LN and %EXP are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types may be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%LOG (numeric-argument)

%LOG accepts a numeric argument and returns the common logarithm (base 10) of the argument. %LOG and %ANTILOG are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types may be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%MAX (numeric-or-string-or-date/time-argument ...)

%MAX accepts one or more character-string, DBCS-string, numeric, or date/time arguments, and returns the largest value from the list. Date/time arguments are arguments of type *DATE, *TIME, or *TIMESTP, or arguments that are date, time, or timestamp durations. String arguments must be no longer than 256 bytes.

If only one argument is specified, this function returns the maximum value of its argument for the group of records defined on the GRPFLD parameter, and the returned value has the same attributes as the argument. If no records are selected, the result is the null value. If the single

argument is a date duration, time duration, or timestamp duration, then the returned value is a packed decimal number (*DEC) with 8 digits and 0 precision (date duration), 6 digits and 0 precision (time duration), or 20 digits and 6 precision (timestamp duration). When a single argument is used, it must be a field name or an expression (not a literal). %MAX with only one argument is an aggregate function that is used for a nongrouping field in a query that uses the grouping function.

If multiple arguments are specified, %MAX returns the maximum value of all the arguments. All arguments must be either character-string, DBCS-string, numeric, or date/time values. This function calculates the maximum value of the first two arguments, and then continues to determine the maximum value of the previous result and the next successive argument. The final result is determined according to the following value conversion rules.

If an argument has different attributes than the previous result, the two values are converted to identical attributes and the operation continues. This conversion uses packed decimal if both values are fixed-point numeric values, or floating-point if either value is floating-point. The conversion for fixed-point numeric values aligns the decimal points and pads the values with zeros. Numeric type changes may truncate fractional digits if more than 31 total digits are required for fixed-point numbers, or drop some of the least significant digits if more than 15 total digits are required for floating-point numbers. Character values are changed by padding the shorter field with blanks.

%MICSEC (date/time-argument)

%MICSEC accepts a date/time argument and returns the microsecond part of the value. The date/time argument can be a timestamp (field or literal), a timestamp duration (field or literal), a character field that contains the external form of a timestamp, or a numeric field or literal. The returned value is of type *BIN4.

A numeric field argument must be defined as packed decimal (*DEC) with 20 digits and 6 precision for timestamp duration. A numeric constant argument must be 14 digits followed by a decimal point and 6 digits.

%MIN (numeric-or-string-or-date/time-argument ...)

%MIN accepts one or more character-string, DBCS-string, numeric, or date/time arguments, and returns the smallest value from the list. Date/time arguments are arguments of type *DATE, *TIME, or *TIMESTP, or arguments that are date, time, or timestamp durations. String arguments must be no longer than 256 bytes.

If only one argument is specified, this function returns the minimum value of its argument for the group of records defined on the GRPFLD parameter, and the returned value has the same attributes as the argument. If no records are selected, the result is the null value. If the single argument is a date duration, time duration, or timestamp duration, then the returned value is a packed decimal number (*DEC) with 8 digits and 0 precision (date duration), 6 digits and 0 precision (time duration), or 20 digits and 6 precision (timestamp duration). When a single argument is used, it must be a field name or an expression (not a literal). %MIN with only one argument is an aggregate function that is used for a nongrouping field in a query that uses the grouping function.

If multiple arguments are specified, %MIN returns the minimum value of all the arguments. All arguments must be either character-string, DBCS-string, numeric, or date/time values. This function calculates the minimum value of the first two arguments, and then continues to determine the minimum value of the previous result and the next successive argument. The final result is determined by the value change rules described below.

If an argument has different attributes than the previous one, the two values are changed to identical attributes and the operation continues. This change uses packed decimal numbers if both values are fixed-point numeric values, or floating-point numbers if either value is a floating-point number. The change for fixed-point numeric values aligns the decimal points and pads with zeros. Numeric type change may truncate fractional digits if more than 31 total digits are required for

fixed-point numbers, or may drop some of the least significant digits if more than 15 total digits are required for floating-point numbers. Character values are changed by padding the shorter field with blanks.

%MINUTE (date/time-argument)

%MINUTE accepts a date/time argument and returns the minute part of the value. The date/time argument can be a time or timestamp field, a time duration or timestamp duration (either field or literal), or a numeric field or literal. The returned value is of type *BIN4.

A numeric field argument must be defined as packed decimal (*DEC) with 6 digits and 0 precision for time duration or packed decimal (*DEC) with 20 digits and 6 precision for timestamp duration. A numeric constant argument must have 6 digits followed by a decimal point, or 14 digits followed by a decimal point and 6 digits.

%MONTH (date/time-argument)

%MONTH accepts a date/time argument and returns the month part of the value. The date/time argument can be a date or timestamp field, a date duration or timestamp duration (field or literal), or a numeric field or literal. The returned value is of type *BIN4.

A numeric field argument must be defined as packed decimal (*DEC) with 8 digits and 0 precision for date duration or packed decimal (*DEC) with 20 digits and 6 precision for timestamp duration. A numeric constant argument must have 8 digits followed by a decimal point, or 14 digits followed by a decimal point and 6 digits.

%NODENAME (integer-argument)

%NODENAME accepts an integer-argument which is used to identify a file that has been specified on the FILE parameter. The argument must be greater than 0 and less than or equal to the number of files specified on the file parameter. The %NODENAME function returns the RDB name for the record retrieved. The returned value is of type *VCHAR of length 18.

Find the node name for every record of the EMPLOYEE table.

```
Example:
OPNQRYF
  FILE((CORPDATA/EMPLOYEE))
  FORMAT(FNAME)
  MAPFLD((NODENAME '%NODENAME(1)'))
```

Join the EMPLOYEE and DEPARTMENT tables, select the employee number (EMPNO) and determine the node from which each record involved in the join originated.

```
Example:
OPNQRYF
  FILE((CORPDATA/EMPLOYEE) (CORPDATA/DEPARTMENT))
  FORMAT(FNAME)
  JFLD((EMPLOYEE/DEPTNO DEPARTMENT/DEPTNO *EQ))
  MAPFLD((EMPNO 'EMPLOYEE/EMPNO'
           (NODENAME1 '%NODENAME(1)')
           (NODENAME1 '%NODENAME(2)'))
```

Join the EMPLOYEE and DEPARTMENT tables, select all records of the result where the records of the two tables are on the same node.

```
Example:
OPNQRYF
  FILE((CORPDATA/EMPLOYEE) (CORPDATA/DEPARTMENT))
  FORMAT(FNAME)
  JFLD((1/NODENAME1 2/NODENAME2 *EQ))
  MAPFLD((NODENAME1 '%NODENAME(1)')
          (NODENAME2 '%NODENAME(2)'))
```

%NODENUMBER (integer-argument)

%NODENUMBER accepts an integer-argument which is used to identify a file that has been specified on the FILE parameter. The argument must be greater than zero and less than or equal

to the number of files specified on the file parameter. The %NODENUMBER function returns a 4-byte binary number (*BIN4) with 10 total decimal digits and no fraction digits. The returned value will be the node number of the record selected.

If the argument identifies a non-distributed file, the value zero is returned.

For OPNQRYF the node number from the secondary file where the outer or exception join is performed will be returned.

If CORPDATA.EMPLOYEE was a distributed file, then the node number for each record and the employee name would be returned.

```
Example:
OPNQRYF
  FILE((CORPDATA/EMPLOYEE))
  FORMAT(FNAME)
  MAPFLD((NODENAME '%NODENUMBER(1)')
         (LNAME '1/LASTNAME'))
```

%NONNULL (argument ...)

%NONNULL accepts a list of two or more arguments and returns the first non-null value from the list. The items in the argument list can be fields or literal values of any type. The type of the returned value is that of the item selected from the list.

```
Example:
OPNQRYF
  FILE(library/file)
  QRYSLT('%NONNULL(fld1 fld2 0) > 0')
```

The above example selects records from the file where either field FLD1 or field FLD2 contains a non-null value that is greater than zero. If both FLD1 and FLD2 were null, the %NONNULL function specified in this example would return '0' because of the constant '0' passed as the third argument. If any field is DBCS-graphic, all fields must be DBCS-graphic.

%NOT (string-argument)

%NOT accepts a character or hexadecimal string argument and returns a string that is the bit-wise 'NOT' (logical not) of the argument. The returned value is a string of type *HEX with the same length as the argument.

%OR (string-argument ...)

%OR accepts two or more character-string arguments and returns a string that is the bit-wise 'OR' (logical inclusive or) of the arguments. This function takes the first argument string, ORs it with the next string, and then continues to OR each successive argument with the previous result. If an argument is encountered that is shorter than the previous result, it is padded with blanks. The final result is a string with the same length as the longest argument. If any of the arguments are variable-length, the maximum length is used as the length of the argument.

%PARTITION (integer-argument)

%PARTITION accepts an integer-argument which is used to identify a file that has been specified on the FILE parameter. The argument must be greater than 0 and less than or equal to the number of files specified on the file parameter. The partition function returns a 4-byte binary number (*BIN4) with 10 total decimal digits and no fraction digits. The returned value will be the partition number of the record.

If the argument identifies a non-distributed file then a value of zero will be returned.

Find the PARTITION number for every row of the EMPLOYEE table. This could be used to determine if there is data skew.

```
Example:
OPNQRYF FILE((CORPDATA/EMPLOYEE))
  FORMAT(FNAME)
  MAPFLD((PART1 '%PARTITION(1)'))
```

Select the employee number (EMPNO) from the EMPLOYEE table for all records where the partition number is equal to 100.

```
Example:
OPNQRYF
FILE((EMPLOYEE))
QRYSLT('%PARTITION(1) *EQ 100')
```

Join the EMPLOYEE and DEPARTMENT tables, select all records of the result where the records of the two tables have the same partition number

```
Example:
OPNQRYF
FILE((CORPDATA/EMPLOYEE) (CORPDATA/DEPARTMENT))
FORMAT(FNAME)
JFLD((1/PART1 2/PART2 *EQ))
MAPFLD((PART1 '%PARTITION(1)')
(PART2 '%PARTITION(2)'))
```

%SECOND (date/time-argument)

%SECOND accepts a date/time argument and returns the seconds part of the value. The date/time argument can be a time or timestamp field, a time duration or timestamp duration (either field or literal), or a numeric field or literal. The returned value is of type *BIN4.

A numeric field argument must be defined as packed decimal (*DEC) with 6 digits and 0 precision for time duration or packed decimal (*DEC) with 20 digits and 6 precision for timestamp duration. A numeric constant argument must have 6 digits followed by a decimal point, or 14 digits followed by a decimal point and 6 digits.

%SIN (numeric-argument)

%SIN accepts a numeric argument and returns the sine of the argument. The argument must be specified in radians. %SIN and %ASIN are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%SINH (numeric-argument)

%SINH accepts a numeric argument and returns the hyperbolic sine of the argument. The argument must be specified in radians.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%SQRT (numeric-argument)

%SQRT accepts a numeric argument and returns the square root of the argument.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%SST (string-argument start-position-expression <length-expression>)

%SST and %SUBSTRING accept a character, hexadecimal, DBCS, or graphic string, a starting position expression, and an optional length expression as arguments. They return a substring of the string argument that is of the same type and CCSID as the string argument and has length equal to the value specified by the length-expression.

Single-byte substringing is done when these functions (%SST and %SUBSTRING) are used for DBCS data. The shift-out and shift-in characters may be lost, which produces unusual results. The result of the DBCS substring operation is the DBCS-open type.

The string argument can be a fixed- or variable-length character, hexadecimal, DBCS, or graphic field or an expression which evaluates to a fixed- or variable-length character, hexadecimal, DBCS, or graphic string.

The values derived from expressions for the second and third arguments must be valid integers. The second argument must have a value between 1 and the length attribute (or maximum length of a variable-length field) of the first argument, and the third argument must have a value between 1 and the length attribute (or maximum length of a variable-length field) of the first argument.

If an argument is DBCS-graphic, the second and third arguments must also be specified as DBCS-graphic characters, not bytes.

If an expression is given for the second or third arguments, the expression must be enclosed in parentheses.

If the expressions evaluate to variable-length results, no validation of the range of these expressions is guaranteed and errors may occur during input/output processing.

The maximum value allowed for the third argument (length) is 32766 except for DBCS-graphic, which is 16383. However, if the third operand is represented by an expression, this causes the result to be variable-length. Thus, the value of the expression cannot exceed 32740 except for DBCS-graphic, which cannot exceed 16370.

The user can omit the third argument. If the third argument is not specified and the first argument is:

- fixed-length, the default value for the third argument is `LENGTH(argument_1) - value_for_argument_2 + 1`
- variable-length, the default value for the third argument is the maximum of 0 and `LENGTH(argument_1) - value_for_argument_2 + 1`
- variable-length with a length less than the value for `argument_2`, the default value for the third argument is zero and the result is the empty string.

Example:

```
OPNQRYF
  FILE(library/file)
  QRYSLT('field1 =
  %SST(field2 (numfld1+3)
  (numfld1+numfld2))')
```

%STDDEV (numeric-argument)

%STDDEV accepts a numeric argument and returns the standard deviation of its argument for the group of records defined by the `GRPFLD` parameter. The argument must be a field name or an expression (not a literal). If no records are selected, the result is the null value. Otherwise, the returned value is a double-precision floating-point number (*FLT8). **%STDDEV** is an aggregate function that is used for a nongrouping field in a query that uses the grouping function.

%STRIP(string-argument <strip-character> <strip-function>)

%STRIP accepts a character-, DBCS-, or graphic- string argument, an optional strip character, and an optional strip function as arguments. It returns a result string with the strip character removed from the string argument as specified by the strip function.

The string argument can be a literal, a fixed or variable-length character, hexadecimal, DBCS, or graphic field, or an expression which evaluates to a fixed- or variable-length character, hexadecimal, DBCS, or graphic string.

The strip character must be a single character, enclosed in apostrophes, with a data type compatible to the source string. The default is a single SBCS space for character data, DBCS-open, and DBCS-either, a single DBCS space for DBCS-only data, and a single graphic space for graphic data.

The strip function can be one of three functions:

***LEAD**

Remove leading strip character(s)

***TRAIL**

Remove trailing strip character(s)

***BOTH**

Remove both leading and trailing strip character(s)

The default strip function is *BOTH.

The return value is a variable-length string with the same type, CCSID, and maximum length as the string argument. If the source string or strip character is null, the result is null.

Example:

```
OPNQRYF
FILE(library/file)
QRYSLT('%STRIP(fld '.' *TRAIL) = 'Mr')
```

%SUBSTRING (string-field-name start-position length)

%SUBSTRING performs the same operation as %SST. See the %SST description for more information.

%SUM (numeric-argument)

%SUM accepts a numeric argument and returns the sum of all the values for its argument in the group of records defined on the GRPFLD parameter and must be enclosed in parentheses. The argument must be a field name or an expression (not a literal).

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. If no records are selected, the result is the null value. Otherwise,

- If the argument is floating-point number, the returned value is a double-precision floating-point number (*FLT8).
- If the argument is a binary number with zero-precision, the returned value is *BIN4.
- If the argument is a binary number with nonzero precision or a fixed-point number, the returned value is a packed decimal number (*DEC) with 31 total digits and as many fractional digits as the argument.
- If the argument is of type date duration, time duration, or timestamp duration, the returned value is a double-precision floating-point number (*FLT8).

%SUM is an aggregate function that is used for a nongrouping field in a query that uses the grouping function.

%TAN (numeric-argument)

%TAN accepts a numeric argument and returns the tangent of the argument. The argument must be specified in radians. %TAN and %ATAN are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The return value is a double-precision floating-point number (*FLT8).

%TANH (numeric-argument)

%TAN accepts a numeric argument and returns the hyperbolic tangent of the argument. The argument must be specified in radians. %TANH and %ATANH are inverse operations.

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. Arguments of these types can be specified either as fields or literal values. The returned value is a double-precision floating-point number (*FLT8).

%TIME (date/time-argument)

%TIME accepts a date/time argument and returns a time. The date/time argument can be a time or timestamp field, a character or hexadecimal field containing the external form of a time, or a time literal. The returned value is of type *TIME.

%TIMESTP (date/time-argument date/time-argument)

%TIMESTP accepts one or two date/time arguments and returns a timestamp.

- If only one date/time argument is specified, it must be a timestamp (field or literal), or a character or hexadecimal field containing the external form of a timestamp.
- If both arguments are specified,
 1. The first date/time argument must be a date (field or literal), or a character or hexadecimal field containing the external form of a date.
 2. The second date/time argument must be a time (field or literal), or a character or hexadecimal field containing the external form of a time.

The returned value is of type *TIMESTP.

%USER

%USER does not support any arguments. It returns the user profile name of the job in which the query is running. The returned value is of type variable-length character (*VCHAR) with a maximum length of 18.

Example:

```
OPNQRYF
  FILE(library/file)
  QRYSLT('field = %USER')
```

%VAR (numeric-argument)

%VAR accepts a numeric argument and returns the variance of its argument for the group of records defined by the GRPFLD parameter. The argument must be a field name or an expression (not a literal).

The following argument types are treated as numeric values: date duration, time duration, and timestamp duration. If no records are selected, the result is the null value. Otherwise, the returned value is a double-precision floating-point number (*FLT8). %VAR is an aggregate function that is used for a nongrouping field in a query that uses the grouping function.

%XLATE (string-argument qualified-table)

%XLATE accepts a character-string argument and the name of a table object (*TBL), and returns a string that is the value of the first argument translated by using the contents of the table. The returned value is a string with the same length and CCSID as the first argument.

The second argument must be a simple or qualified table object name. If no library name is specified, *LIBL is used to find the table.

%XOR (string-argument...)

%XOR accepts two or more character-string arguments and returns a string that is the bit-wise 'XOR' (logical exclusive or) of the arguments. This function takes the first argument string, XORs it with the next string, and then continues to XOR each successive argument with the previous result. If an argument is encountered that is longer than the previous result, the previous result is padded with blanks before the XOR operation. If any of the arguments is variable-length, the maximum length is used as the length of the argument. The final result is a string of type *HEX with the same length as the longest argument.

%YEAR

%YEAR accepts a date/time argument and returns the year part of the value. The date/time argument can be a date or timestamp field, a date duration or timestamp duration (field or literal), or a numeric field or literal. The returned value is of type *BIN4.

A numeric field argument must be defined as packed decimal (*DEC) with 8 digits and 0 precision for date duration or packed decimal (*DEC) with 20 digits and 6 precision for timestamp duration. A numeric constant argument must have 8 digits followed by a decimal point, or 14 digits followed by a decimal point and 6 digits.

Restricted Built-in Functions

The following built-in function is supported only as the second operand of the 'equal' or 'not-equal' relational operators specified on the QRYSLT or GRPSLT parameter.

%NULL

%NULL accepts no arguments. It is used to select or omit records based on whether or not a field in the record contains a null value.

Example:

```
OPNQRYF
FILE(library/file)
QRYSLT('charfld = %NULL')
```

This query would select all the records where 'charfld' contains the null value.

The following three built-in functions are supported only as the second operand of the 'equal' relational operator specified on the QRYSLT or GRPSLT parameter.

%RANGE (low-value high-value)

%RANGE is used to identify the lower and upper boundaries for the value of a field or expression. %RANGE must be specified as the right side of a relation whose operator is equal. The low-value and high-value argument must be field names, character strings, or numeric literals, to match the type of field or expression specified as left side of the relation. For example, to select only records where the numeric field NBRFLD has a value ranging from 10 through 20, specify:

```
'nbrfld = %RANGE(10 20)'
```

If the low-value argument is greater than the high-value argument, the relation produces a logical value of 'false'.

%VALUES (allowed-value...)

%VALUES is used to identify a list of allowed values for a field or expression. %VALUES must be specified as the right side of a relation whose operator is equal. The allowed-value arguments must be character string or numeric literals, to match the type of the field or expression specified as the left side of the relation. For example, to select only records where the second character of field CHARFLD has a value that is one of the values 'A', 'E', 'I', 'O', or 'U', specify the following:

```
'%SST(charfld 2 1) =
%VALUES('A' 'E' 'I' 'O' 'U')
```

%WLDCRD ("pattern-string" "wild-characters")

%WLDCRD is used to specify a pattern that performs a wildcard scan of the character or hexadecimal field or string expression (except for expressions made up of a single character-string literal) that must be specified as the left side of the relation. %WLDCRD must be specified as the right side of a relation whose operator is equal. The pattern-string argument must be a character-string, DBCS, or graphic literal, to match the left side of the relation. The wild-characters argument is an optional parameter that specifies what 'wildcard' characters are used in the pattern-string.

If specified for character data only (no DBCS data), the wild-characters argument must be a character-string literal of exactly two characters. The first character is the value that matches any single character in the search string. The second character is the value that matches a substring of any zero or more characters. The two characters must not be the same, but there is no requirement that either character appear in the pattern-string. If the wild-characters argument is omitted, the default is for an underline ('_') to match any single character and an asterisk ('*') to match a substring of any zero or more characters.

If the wild-characters argument is specified for DBCS data only (no character data), the argument must be a double-byte character-string literal of exactly two double-byte characters. The first double-byte character is the value that will match any one double-byte character in the search string. The second double-byte character is the value that will match a substring of any zero or more characters. The two double-byte characters must not be the same, but there is no requirement that either character appear in the pattern string. If the wild-characters argument is omitted, the default is for a DBCS underline to match any one double-byte character and a DBCS asterisk to match a substring of any zero or more double-byte characters.

If the wild-characters argument is specified for both character and DBCS data, in addition to the previous rules, the argument must first contain a single-byte character-string literal (two single-byte characters), then a double-byte character string (two double-byte characters).

In this case, the first character matches any single-byte character in the character string, the second character matches a substring of any number of single-byte or double-byte characters. The first double-byte character matches any double-byte character in the character string. The second double-byte character matches a substring of any number of single-byte or double-byte characters.

The following example selects only records where the character field CHARFLD contains a 'T', followed by any two characters and an 'E', appearing anywhere in the field.

```
'charfld = %WLDCRD('*T__E*')
```

Note:

The asterisks at the start and end of the pattern-string are required to allow the 'T' and 'E' to appear somewhere other than the first and last positions in the field:

To select only records where the character field CHARFLD starts with the string 'ABC', followed by one or more other characters and then followed by the string 'XYZ' (but not necessarily at the end of the field), specify the following:

```
'charfld = %WLDCRD(''ABC_*XYZ*')
```

To select only records where the second character of field CHARFLD is an asterisk ('*'), the last character is an underline ('_'), and the letter 'M' appears somewhere in between, specify the following:

```
'charfld = %WLDCRD(''##.M._'' '#.')
```

Error messages for OPNQRYF

*ESCAPE Messages

CPF2115

Object &1 in &2 type *&3 damaged.

CPF2169

Job's sort sequence information not available.

CPF2619

Table &1 not found.

CPF3BCC

Language identifier &1 not valid.

CPF3BC6

Sort sequence &1 not valid.

CPF3BC7

CCSID &1 outside of valid range.

CPF3BC8

Conversion from CCSID &1 to CCSID &2 is not supported.

CPF3BC9

Conversion from CCSID &1 to CCSID &2 is not defined.

CPF3BDD

Sort sequence &1 not valid for UCS2 data.

CPF3FC0

Language identifier is not valid.

CPF4174
OPNID(&4) for file &1 already exists.

CPF8133
Table &4 in &9 damaged.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9813
Record format &3 in file &1 not found.

CPF9815
Member &5 file &2 in library &3 not found.

CPF9820
Not authorized to use library &1.

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

CPF9826
Cannot allocate file &2.

CPF9830
Cannot assign library &1.

CPF9899
Error occurred during processing of command.

***STATUS Messages**

CPI4011
Query running. &2 records selected, &1 processed.

CPI4301
Query running.

CPI4302
Query running. Building access path for file &1 in &2.

CPI4303
Query running. Creating copy of file &1 in &2.

CPI4304
Query running. &1 records selected. Selection complete.

CPI4305

Query running. Sorting copy of file *N in *N.

CPI4306

Query running. Building access path from file &1 in &2.

CPI4307

Query running. Building hash table from file &1 in &2.



ORDSPTPTF (Order Supported Product PTFs) Command Description

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

ORDSPTPTF Command syntax diagram

Purpose

The Order Supported Product PTFs (ORDSPTPTF) command orders program temporary fixes (PTFs) for supported products. This is done by comparing the PTFs on the system with the PTFs available from IBM service support, and then ordering the PTFs that are:

1. Not loaded
2. Not applied
3. Not on order
4. Not in a PTF save file (SAVF) in the QGPL library

The PTF order request created by this command is sent to IBM service support.

This command detects any PTFs on the system that IBM now considers defective. This command also detects PTFs not on the system that IBM considers High-Impact Pervasive (HIPER). In either case, a message indicating a PTF number and a recommended action is sent to the service provider message queue.

Required Parameter

DELIVERY

Specifies how PTFs are delivered.

***ANY:** Any method of delivery is acceptable. If the order that is sent to IBM service support is small enough, the PTF order arrives by way of electronic customer support. If the PTF order is larger than the acceptable electronic customer support limit, PTFs arrive by way of tape.

***LINKONLY:** Only delivery by way of electronic customer support is acceptable. If the PTF order is larger than the acceptable electronic customer support limit, a message is sent indicating that the PTF delivery failed.

Optional Parameter

LICPGM

Specifies which licensed programs are having PTFs ordered for them.

***ALL:** PTFs are ordered for all licensed programs.

licensed-program: Specify the licensed program that is having PTFs ordered for it.

Examples for ORDSPTPTF

Example 1: Ordering PTFs for All Supported Products

```
ORDSPTPTF DELIVERY(*ANY)
```

This command orders PTFs for all supported products. If the order is small enough, the PTFs arrive by way of electronic customer support; otherwise, they arrive by tape.

Example 2: Ordering PTFs for a Licensed Program

```
ORDSPTPTF DELIVERY(*LINKONLY) LICPGM(5722SS1)
```

This command orders PTFs for the licensed program named 5722SS1 (OS/400). The PTFs arrive by way of electronic customer support.

Error messages for ORDSPTPTF

*ESCAPE Messages

CPF8C07

A parameter is not valid.



OVRDBF (Override with Database File) Command Description

OVRDBF Command syntax diagram

Purpose

The Override with Database File (OVRDBF) command is used to:

- Override (replace) the file named in the program
- Override certain parameters of a file that are used by the program
- Override the file named in the program and override certain parameters of the file processed

Parameters overridden by this command are specified in the file description, in the program, or in other previously issued file override commands. The OVRDBF command applies to physical files, logical files, and distributed data management (DDM) files.

To override (replace) a file named in the program, specify the name of that file in the FILE parameter, and specify the name of the file that overrides it (the file to be processed by the program) in the TOFILE parameter. The other parameters of this command can be used to override parameter values contained in the file description of the overriding file.

To override only certain parameters of the file named in the program, instead of replacing the entire file, specify the name of the file in the FILE parameter and specify the *FILE value for the TOFILE parameter. Then use the other parameters of this command to override specific parameters of the file. Parameters that are not specified do not affect parameters specified in the file description, in the program, or in other previously issued file override commands.

Restrictions:

1. In a multithreaded job, this command may only be issued from the initial thread.
2. In a multithreaded job, only overrides scoped to the job or an ILE activation group will affect opens performed in a secondary thread.

More information on overriding files is in the File Management topic in the Information Center and the Database Programming topic in the Information Center.

Required Parameter

FILE Specifies the name of the file being used by the program to which this override command is applied. If TOFILE(*FILE) is specified, a display device file must be specified. Otherwise, any device file or database file can be specified.

Optional Parameters

TOFILE

Specifies the qualified name of the database file that is used instead of the file specified in the FILE parameter or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified on this OVRDBF command override the same parameters specified in the database file, in the program, or in other previously issued OVRDBF commands.

***FILE:** The database file named in the FILE parameter has some of its parameters overridden by values specified in this command.

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

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

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

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

database-file-name: Specify the name of the database file that is used instead of the file specified in the FILE parameter.

MBR Specifies the members used within the database file. This parameter is not valid for DDM files that reference remote systems other than the System/38 or the iSeries 400.

***FIRST:** The first member in the database file is used.

***LAST:** The last member of the specified physical file is used.

***ALL:** All members in the file are processed sequentially. All members are opened with the same override parameters as the first member. While overrides issued prior to the open operation of the first member are processed, overrides or delete overrides issued following the open operation of the first member are not processed. EOFDLY, FMTSLR, INHWRT, or the POSITION parameter cannot be specified if MBR(*ALL) has been specified on a previously issued OVRDBF command that is still in effect for this file. An escape message is sent if any of the mutually exclusive parameters are specified.

member-name: Specify the member name that overrides (at file open time) the member name specified in the using program, or in other called OVRDBF commands. If the member name is not specified, and a TOFILE parameter other than *FILE has been specified, the first member in the file is used.

POSITION

Specifies the starting position for retrieving records from the database file. The first record to get can be at the beginning (*START) or at the end (*END) of the file, the nth record in the file (*RRN), or the record indicated by a key field value and one of the key-search values (*KEY, *KEYA, *KEYAE, *KEYB, or *KEYBE). This parameter overrides the value specified in the program, or in other called OVRDBF commands.

Note:

This parameter cannot be specified if MBR(*ALL) was specified in a previously issued OVRDBF command that is still in effect for this file.

***NONE:** No special positioning is required. The first I/O operation indicates the record that is retrieved.

***START:** The starting position is the first record in the file. If a read-previous is specified in the program, an end-of-file condition occurs.

***END:** The starting position is the last record in the file. When the next record is retrieved, an end-of-file condition is reached. If a read-previous is requested, the last record of the file is retrieved.

Element 1: Relative Record Number

***RRN** *relative-record-number*: Specify the number of the relative record (its position from the beginning of the file), preceded by the value *RRN, that is retrieved first. For example, POSITION(*RRN 480) specifies that record number 480 is retrieved next. If a read-previous is requested, the 479th record in the file is retrieved.

Element 2: Key-Search Values

The first record that is retrieved is identified by the specified key-operation, number-of-fields, record-format-name, and key-value. If a record that matches these values does not exist, an error message is sent.

Specify one of the following key-search types:

***KEYB** (*key-before*): A record that precedes the record identified by the remaining search values (number-of-fields, record-format-name, and key-value) is the first record retrieved.

***KEYBE** (*key-before or equal*): The record identified by the search values is the first record retrieved. If no record matches those values, the record is selected that matches the largest previous value.

***KEY** (*key-equal*): The record identified by the search values is the first record retrieved. If a read-previous is specified in the program, the preceding record is retrieved.

***KEYAE** (*key-after or equal*): The record identified by the search values is the first record retrieved. If no records matches those values, the record is selected with the next highest value.

***KEYA** (*key-after*): A record that follows the record identified by the remaining search values (number-of-fields, record-format-name, and key-value) is the first record retrieved.

Specify the remaining search values as follows:

Element 3: Number of Fields

number-of-fields: Specify the number of key fields to use in the search. The number of fields specified in this parameter does not have to be the same as the actual number of fields in each key for the file. For example, if POSITION(*KEY 1 FMT1 A) is specified, the first record in the file

format FMT1 that has a first key field value of A is retrieved. If a number of fields of zero are specified, the search is based on all key fields. If zero is used, the key value contains the maximum key size.

Element 4: Name of Record Format

record-format-name: Specify the name of the record format in the database file that contains the key value specified. If no record format name is specified (*N), all record formats are searched for the first record that matches the other search values.

Element 5: Key Value

key-value: Specify the first record retrieved. This value is specified as a quoted character string for character or positive zoned decimal formats, or is specified in hexadecimal form at (x'value'). You can specify up to 2000 characters in the character string.

For example, POSITION(*KEY 1 FMT2 X'123F') specifies that the system searches for a record from the record format FMT2, that a single key field is used in the search (even though the key value may have more key fields), and that the record contains the hexadecimal value 123F (the hexadecimal equivalent of packed decimal value 123).

POSITION(*KEYB 0 *N X'123F') specifies that a record of any format is retrieved next (its key value must precede the record identified by key value X'123F').

If a key is specified that contains more than one field, the key must be coded to match the definition of the key in the file. If the definition is for a key other than a character or signed decimal key, the key must be coded in hexadecimal form.

For example, suppose the key definition has the following key fields:

- Character field (6A)
- Packed numeric field (5P 2)
- Signed numeric field (2S 0)

A character string the length of the entire key (6+3+2 in this example) can be specified on the POSITION parameter. POSITION(*KEY 3 YOURFMT X'E6D9C5D5C3C812345FF9F9') specifies that the system searches for the record in format YOURFMT, a key containing three fields is used in the search, and the record contains the hexadecimal value E6D9C5D5C3C812345FF9F9. The hexadecimal value corresponds to the following desired key values:

- Hexadecimal value E6D9C5D5C3C8 corresponds to the character field key value WRENCH.
- Hexadecimal value 12345F corresponds to the packed numeric field value +123.45.
- Hexadecimal value F9F9 corresponds to the signed numeric field value 99.


The Distributed Data Management topic in the Information Center has more information on the effects of using the POSITION parameter with DDM files.

RCDFMTLCK

Specifies the lock state of the named record format while it is used by the program. The lock state indicates how the data associated with each format is locked. The following chart shows the lock states that are specified for each record format and the operations allowed to other programs when the lock is in effect:

Lock State	Definition	Other Program Operations
*SHRRD	Shared read	Read and update allowed
*SHRNUP	Shared read, no update	Read allowed, update not allowed
*SHRUPD	Shared update	Read and update allowed

Lock State	Definition	Other Program Operations
*EXCLRD	Exclusive allow read	Read allowed, update not allowed
*EXCL	Exclusive no read	Neither read nor update allowed

An explanation of each lock state is in the CL Programming  book.

For each record format, specify the record format name followed by one lock state value. This parameter overrides the record format locks specified in the program, in other called OVRDBF commands, and the default locks established when the member was created. If the lock state specified for the file in an Allocate Object (ALCOBJ) command is more restrictive than the lock state specified in this parameter, this parameter is ignored. Therefore, this parameter can only impose a more restrictive lock state on a record format than that specified for the file.

FRCRATIO

Specifies the number of insert, delete, or update operations that can occur on records before they are forced into auxiliary (permanent) storage. More information on this parameter is in commonly used parameters. More information on journal management is in the Journal management article in the Information Center.

This parameter overrides the force-write ratio specified in the database file, in the program, or in other previously issued OVRDBF commands.

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

number-of-write-operations-before-force: Specify the number of operations. If a physical file associated with this database file is journaled, specify a larger force-write ratio.

FMTSLR

Specifies the qualified name of a record format selection program that is called when a logical file member contains more than one logical record format. The user-written selector program is called when a record is inserted into the database file and a record format name is not specified in the high-level language program. More information about the use of format selector programs is in the Database Programming topic in the Information Center. This parameter overrides the value specified in the database file and in other previously issued OVRDBF commands.

A program specified as the format selector program cannot be created with USRPRF(*OWNER) specified in the Create CL Program (CRTCLPGM) command.

Notes:

1. This parameter cannot be specified if MBR(*ALL) was specified in a previously issued OVRDBF command that is still in effect for this file.
2. If a less restrictive state than that of the actual file is specified, the value will be ignored and the original value specified in the file will be used.

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 a record format selection program called when a logical file member contains more than one logical record format.

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. If those resources are not allocated within the specified wait time, an error message is sent to the program.

This parameter overrides the wait time specified in the database file, in the program, or in other previously issued OVRDBF commands. More information on this parameter is in commonly used parameters.

***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.

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.

Note:

This parameter overrides the record wait time specified in the database file, specified in the program, or in other previously issued OVRDBF commands. The minimum delay for DDM files is 60 seconds. This value may need to be longer than the delay specified for local database files.

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

***NOMAX**: There is no disconnect limit.

number-of-seconds: Specify the number of seconds that the program waits for the record lock. Valid values range from 1 through 32767 seconds.

NBRRCDS

Specifies the number of records moved as a unit from auxiliary storage to main storage. (The amount of data actually moved is equal to the number of records multiplied by the physical record length, not the logical record length.) Valid values range from 1 through 32767 records. The NBRRCDS parameter is valid for sequential or random processing and is specified only when the data records are physically located in auxiliary storage in the sequence in which they are processed. This parameter overrides the number of records value specified in the program, or in other previously issued OVRDBF commands.

EOFDLY

Specifies the number of seconds to delay when the end-of-file is reached before trying to retrieve additional records. This delay allows other jobs to add records to the file, and have the new records processed without having to restart the job. When the delay time ends, the job is made active, and the database determines whether new records were added. If no new records were added, the job waits for another time delay without informing the application program. When a number of seconds is specified, no end-of-file condition occurs on the given database file until an End Job (ENDJOB) command or forced end of data (FEOD) occurs.

Note:

This parameter cannot be specified if MBR(*ALL) was specified on a previously issued OVRDBF command that is still in effect for this file.

There are several ways to end a job that is waiting for records due to an EOFDLY. They are:

- Write a record to the specified file which is recognized by the application program as a last record. The application program may then do a force end of data (FEOD) to start the end-of-file processing or close the file.
- End the job using the controlled value (ENDJOB OPTION(*CNTRLD)) with a delay time greater than the time specified on the EOFDLY time. The DELAY parameter time specified must allow for the EOFDLY time to run out, plus time to process any new records that may have been added to the file, and any end-of-file processing that is done in the user's application. The end-of-file is set by database, and a normal end-of-file condition occurs after new records are retrieved.
- End the job immediately (ENDJOB OPTION(*IMMED)).
- If the job is interactive, start a system request and end the previous request.

***NONE:** Normal end-of-file processing is done.

number-of-seconds: Specify the number of seconds that the program waits between each attempt to get a record when an end-of-file condition occurs. No end-of-file condition is signaled until end of data is forced, or the job is ended with the *CNTRLD option. Valid values range from 1 through 99999 seconds.

LVLCHK

Specifies whether the record format level identifiers in the program are checked against those in the device file when the file is opened. If so, the record format identifiers in the program must match those in the device file. Because the same record format name can exist in more than one file, each record format is given an internal system identifier when it is created.

Note:

This parameter overrides the value specified in the database file, in the program, or in other previously issued OVRDBF commands issued in this or the following call level. Level checking cannot be done unless the program contains the record format identifiers. This command cannot override level checking from *NO to *YES.

***NO:** The level identifiers are not checked when the file is opened.

EXPCHK

Specifies whether the expiration date of the named member is checked. This date check is valid only on a physical file member. This parameter overrides the value specified in the program, or in other called OVRDBF commands.

***YES:** The expiration date of the physical file member is checked. If the current date is later than the expiration date, an error message is sent to the job, where it is monitored. An escape message is sent to the program.

***NO:** The expiration date is not checked.

INHWRT

Specifies whether the processed records are written, deleted, or changed in the database file. This

parameter tests a program without storing the processed records back in the database. This parameter overrides the INHWRT parameter in other previously issued OVRDBF commands.

Note:

This parameter cannot be specified if MBR(*ALL) was specified on a previously issued OVRDBF command that is still in effect for this file.

***YES:** Processed records are prevented from being written into the database; they are written only to an output device.

***NO:** All new and changed processed records are written into the database, unless the program is in debug mode with UPDPROD(*NO) specified, and the file is in a production library. In that case, an escape message is sent to the program.

SECURE

Specifies whether this file is safe from the effects of previously called file override commands. If SECURE is not specified, processing occurs as if SECURE(*NO) is specified.

***NO:** This file is not protected from the effects of other file overrides; its values can be overridden by the effects of previously called file override commands.

***YES:** This file is protected from the effects of any file override commands previously called.

OVRSCOPE

Specifies the extent of influence (scope) of the override.

***ACTGRPDFN:** The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL:** The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB:** The scope of the override is the job in which the override occurs.

SHARE

Specifies whether the open data path (ODP) for the database file member 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.

Note:

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, 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.

OPNSCOPE

Specifies the extent of influence (scope) of the open operation.

***ACTGRPDFN:** The scope of the open data path (ODP) is determined by the activation group of the program that called the OVRDBF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller. In a multi-threaded job, only those shared opens within the same thread and same activation group can share this ODP.

***JOB:** The scope of the open data path (ODP) is the job in which the open operation occurs. If the job is multi-threaded, only those opens from the same thread can share this ODP.

SEQONLY

Specifies, for database files whose records are processed in sequential order only, whether sequential-only processing is used on the file. This parameter also specifies the number of records transferred as a group to or from the database, if sequential-only processing is used. If a number is not specified, a default number is determined by the system. This parameter is used to improve the performance of programs that process database files in a sequential manner. This parameter overrides the value specified in the program or in other previously issued OVRDBF commands.

For files opened for *input* only in a program, the specified number of records is transferred as a group from the database to an internal data management buffer.

For files opened for *output* only in a program, a group of records is transferred to the database whenever the internal data management buffer receives the specified number of processed records from the program. For output files, sequential-only processing is valid for physical file members and for logical file members that are based on one physical file member only.

If SEQONLY(*YES) is specified, and any of the following conditions are true, the SEQONLY parameter is ignored and a message is issued.

- The program opened the member for output only and SEQONLY(*YES) is specified with the default number of records, and the member opened is either a logical member, a unique keyed physical member, or other access paths are built over the physical member.
- The program opened the member for other than input or output.
- The member opened by the program for output is based on many other members.
- The record length plus the feedback area sum exceeded 32,767 bytes.

Note:

Unpredictable results occur when this parameter is used for alternate index files for DDM on a system other than an iSeries 400.

***NO:** The database file is not restricted to sequential-only processing.

***YES:** The database file uses sequential-only processing. A default value for the number of records transferred as a group is determined by the system based on how the file is used, the type of access path involved, and the file's record length:

- The default is approximately the number of records that fit in an internal buffer of 4K for:
 - All database files opened for input only
 - Physical files opened for output that are only processed in either arrival sequence or in non-unique keyed sequence and that have no logical file members based on them
- The default is 1 record for:

- All logical files opened for output only
- Physical files opened for output only that either have *unique* keyed sequence access paths or have at least one dependent logical file with a keyed sequence access path that does not share the access path of the keyed physical file member

number-of-records: Specify *YES followed by a value (ranging from 1 through 32767) for the number of records transferred between the database and the internal buffer. The user must ensure that the buffer size specified is always available to the program in the storage pool in which the program is running. The file uses sequential-only processing.

While records are in the internal data management buffer, other jobs can make changes to the same records in the database, and the program performing sequential-only input processing does not see the updates. To ensure that no other updating is done to records while they are in the buffer, the Allocate Object (ALCOBJ) command can be used in the program to specify either an *EXCLRD or an *EXCL lock on the file.

If a program performs sequential-only output processing and does not handle output errors (such as duplicate keys and conversion mapping errors) that may occur when the records in the buffer are written to the database, records in the buffer after the first record in error are not written.

If the file is opened for output and the value specified in this parameter is not the same as the force write ratio specified for the file, the value used by the system is the smaller of the two; a message stating which value is changed is sent to the user.

When processing SEQONLY(*YES) for writing records into a database file, feedback information for each record (such as relative record number) is not always changed. If such feedback information is important, specify SEQONLY(*NO) or SEQONLY(*YES 1).

More information on sequence-only database files is in the Database Programming topic in the Information Center.

DSTDTA

Specifies the data retrieval method used for a distributed file. This parameter has no effect if used against a non-distributed file. Other parameters, such as SEQONLY, still affect how the data is retrieved from each system, and this parameter controls how all the data is managed when accessing a distributed file. This parameter overrides the distributed file data retrieval method selected by the system, or specified in other previously issued OVRDBF commands.

***BUFFERED:** In order to achieve the best performance, data from the remote system and the local system may be kept in a buffer until retrieved by the user.

***PROTECTED:** Data can be buffered, but the file is locked to prevent updates by other jobs. This will give the same performance as *BUFFERED, but guarantees current data. While one job is using this option, other jobs will not be able to update the data in the file.

***CURRENT:** Data is not buffered. This option results in fully live data, with maximum concurrency, but without optimal performance.

Examples for OVRDBF

Example 1: Overriding An Existing Member

```
OVRDBF FILE(ORDERSIN) MBR(MONDAY)
```

This command overrides the existing member with member MONDAY. With the override in effect, the member MONDAY will be processed when the file ORDERSIN is opened.

Example 2: Overriding a Share Specification

```
OVRDBF FILE(ORDERSIN) SHARE(*YES)
```

This command overrides the share specification for the file ORDERSIN. Because of this override, any subsequent opens of this file within the routing step share the ODP for the file.

Example 3: Overriding a File, Member and Lock State

```
OVRDBF FILE(INPUT) TOFILE(PAYROLL) MBR(MBR1)
      RCDFMLCK((EMPDATA *EXCL))
```

This command overrides the file, the member, and the lock state of the record format EMPDATA. The override will cause the following to occur when the file INPUT is opened:

- The file PAYROLL will be processed instead of the file INPUT.
- The member MBR1 will be processed instead of the previously specified member.
- The lock *EXCL will be placed on record format EMPDATA instead of the existing lock. (*EXCL prevents another program from using the record format while the override is in effect.)

Additional Considerations

The following characteristics apply to the processing of DDM files on the OVRDBF command.

- All parameters are processed normally when the target system is an iSeries 400.
- When the target system is not an iSeries 400:
 - The following parameters are still valid:

EXPCHK	POSITION	SEQONLY	WAITFILE
INHVRT	RCDFMLCK	SHARE	WAITRCD
LVLCHK	SECURE	TOFILE	

- The FMTSLR parameter, if specified, causes an error when the file opened is a DDM file.
- The FRCRATIO and NBRRCDs parameters, if specified, are ignored.
- The RCDFMLCK parameter, if specified, is valid only if both of the following are true of the remote file used: (1) Only one type of lock state is requested for the remote file, and (2) the record format name in the remote file must be the same as the name of the distributed data management file.
- The TOFILE parameter is always processed on the source system. When a DDM file name is specified on this parameter, the program uses the associated remote file instead of the local database file specified in the program.
- The WAITFILE and WAITRCD parameters have no effect on remote file processing.

The Distributed Data Management topic in the Information Center has examples of how file overrides are applied in DDM.

Error messages for OVRDBF

*ESCAPE Messages

CPF180C

Function &1 not allowed.


OVRDKTF (Override with Diskette File) Command Description

OVRDKTF Command syntax diagram

Purpose

The Override with Diskette File (OVRDKTF) command is used to (1) override (replace) the file named in the program, (2) override certain parameters of a file that is used by the program, or (3) override the file named in the program and override certain parameters of the file processed.

Parameters overridden by this command are specified in the file description, in the program, or in other called file override commands. If a file named in the program is overridden, the name of that file is specified in the FILE parameter and the name of the overriding file (the file processed) is specified in the TOFILE parameter. The OVRDKTF command also specifies parameters to override values contained in the file description of the overriding file. If the file named in the program is not replaced but certain parameters of the file are overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters overridden are then specified by the other parameters of the OVRDKTF command. Parameters that are not specified do not affect parameters specified in the file description, in the program, or in other called file override commands.

More information on overriding files is in the File Management topic in the Information Center, and the Tape and Diskette Device Programming  book.

Required Parameter

FILE Specifies the name of the file being used by the program to which this override command is applied. If TOFILE(*FILE) is specified, a display device file must be specified. Otherwise, any device file or database file can be specified.

Optional Parameters

TOFILE

Specifies the qualified name of the diskette file that is used instead of the file specified in the FILE parameter or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified in this OVRDKTF command override the same parameters specified in the diskette device file in the program, or in other called OVRDKTF commands.

***FILE:** The diskette device file named in the FILE parameter has some of its parameters overridden by values specified in this command.

The name of the diskette 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.

diskette-device-file-name: Specify the name of the diskette device file that is used instead of the overridden file.

DEV Specifies the name of the diskette device used with this diskette device file to perform input/output data operations. The device name of the IBM-supplied diskette device description is QDKT. This parameter is ignored if SPOOL(*YES) is specified for the file when it is opened.

This parameter overrides the value specified in the device file, in the program, or in other called OVRDKTF commands.

device-name: Specify the name of the device that is used with this diskette device file. The device name must already exist on the system as 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.

This parameter overrides the volume identifiers specified in the diskette device file, in the program, or in other called OVRDKTF commands.

***NONE:** The diskette volume identifiers are not specified for this file in this command. They can be specified later before the device file is opened, either in a Override with Diskette File (OVRDKTF) command or a Change Diskette File (CHGDKTF) command, or in the high-level language program. Otherwise, no volume identifier checking is done.

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

LABEL

Specifies the data file label of the data file on diskette that is used with this diskette device file. For input files (diskette input to system), this label specifies the identifier of the file that exists on the diskette. For output files (system output to diskette), the label specifies the identifier of the file that is created on the diskette. More information on this parameter is in commonly used parameters.

This parameter overrides the label specified in the diskette device file, in the program, or in other called OVRDKTF commands.

data-file-label: Specify up to 8 characters for the identifier of the data file used with this diskette device file.

EXCHTYPE

Specifies, for diskette output files only, the exchange type used by the device file when the system is writing diskette data. More information on this parameter is in commonly used parameters.

This parameter overrides the value specified in the device file, in the program, or in other called OVRDKTF commands.

***STD:** The basic exchange format is used for a type 1 or a type 2 diskette. The H exchange type is used for a type 2D diskette.

***BASIC:** The basic exchange type is used.

***H:** The H exchange type is used.

***I:** The I exchange type is used.

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).

This parameter overrides the value specified in the device file, in the program, or in other called OVRDKTF commands.

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

CRTDATE

Specifies the date when the diskette data file was created on the diskette.

Note:

The creation date parameter is valid only for diskette input data files. If the creation date written on the diskette containing the data file does not match the date specified for the device file when it is opened, an error message is sent to the user program.

This parameter overrides the values specified on the device file, in the program, or on other called OVRDKTF commands.

***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. However, the specified date is put in the diskette label in the format yymmdd.

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:

The expiration date overrides the value specified in the device file, in the program, or in other called OVRDKTF commands.

***NONE:** No expiration date for the data file is specified; the file is protected for 1 day. Its protection ends the day after it is created.

***PERM:** The data file is permanently protected. An expiration date of 999999 is assigned.

expiration-date: Specify the expiration date of the data file. The date must be specified in the format defined by the job attributes DATFMT and, if separators are used, DATSEP. However, the specified date is put in the diskette label as yymmdd.

SPOOL

Specifies whether the input or output data for the diskette device file is spooled.


This parameter overrides the spool value specified in the device file, or in other called OVRDKTF commands.

***NO:** The data is not spooled. If this file is opened for input, the data is read directly from the diskette. If this is an output file, the data is written directly to the diskette as it is processed by the program.

Note:

If SPOOL(*NO) is specified, the following parameters in this command are ignored: OUTQ, MAXRCDS, SCHEDULE, HOLD, SAVE, OUTPTY, and USRDTA.

***YES:** The data is spooled. If this file is opened for input, an inline data file having the specified name is processed; otherwise, the next unnamed inline spooled file is processed. More information

on named and unnamed inline files is in the Tape and Diskette Device Programming  book. If this is an output file, the data is spooled for processing by a diskette or print writer.

OUTQ Specifies the qualified name of the output queue used for spooled printer files that specify OUTQ(*JOB). This change does not affect files already created in active jobs or files in completed jobs in which the files were spooled.

This parameter overrides the output queue name specified in the device file, or in other called OVRDKTF commands.

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 to which the output data is spooled. The IBM-supplied output queue that is used by the diskette file is the QDKT output queue, stored in the QGPL library.

MAXRCDS

Specifies, for spooled files only, the maximum number of records in the spooled file for spooled jobs using this diskette device file.

This parameter overrides the value specified in the diskette device file, or in other called OVRDKTF commands.

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

maximum-records: Specify the maximum number of diskette records that are in the spooled file. Valid values range from 1 through 500000.

SCHEDULE

Specifies, for spooled output only, when the spooled file is available to a writer.

This parameter overrides the scheduling value specified in the device file, or in other called OVRDKTF commands.

***JOBEND:** The spooled file is made available to the writer only after the entire job is completed.

***FILEEND:** The spooled file is made available to the writer as soon as the file is closed in the program.

***IMMED:** The spooled file is made available to the writer as soon as the file is opened in the program.

HOLD Specifies, for spooled output only, whether the spooled file is held. The spooled file can be released by using the Release Spooled File (RLSSPLF) command.

Note: This parameter overrides the hold value specified in the diskette device file, or in other called OVRDKTF commands.

***NO:** The spooled printer file is not held by the output queue. The spooled output is available to a writer based on the SCHEDULE parameter value.

***YES:** The spooled file is held until released by the Release Spool File (RLSSPLF) command.

SAVE Specifies, for spooled output only, whether the spooled file is saved (left on the output queue) after the output has been produced.

Note: This parameter overrides the save value specified in the diskette device file, or in other called OVRDKTF commands.

***NO:** The spooled file data is not saved on the output queue after it has been produced.

***YES:** The spooled file data is saved on the output queue until the file is deleted.

OUTPTY

Specifies the output priority for spooled output files that are produced by this job. The highest priority is 1 and the lowest priority is 9. More information on this parameter is in commonly used parameters.

***JOB:** The output priority associated with the job that created the spooled file is used.

output-priority: Specify the output priority. Valid values range from 1 (high priority) through 9 (low priority).

USRDTA

Specifies, for spooled output only, the user-specified data that identifies the file.

***BLANK:** Ten blanks are used as the user data.

user-data: Specify up to 10 characters of text.

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.

This parameter overrides the wait time specified in the device file, in the program, or in other called OVRDKTF commands.

***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 diskette file 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.

SECURE

Specifies whether this file is safe from the effects of previously called file override commands. If SECURE is not specified, processing occurs as if SECURE(*NO) is specified.

***NO:** This file is not protected from the effects of other file overrides; its values can be overridden by the effects of previously called file override commands.

***YES:** This file is protected from the effects of any file override commands previously called.

OVRSCOPE

Specifies the extent of influence (scope) of the override.

***ACTGRPDFN:** The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL:** The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB:** The scope of the override is the job in which the override occurs.

SHARE

Specifies whether the open data path (ODP) for the diskette 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.

This parameter also overrides the value specified in other called OVRDKTF commands.

***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.

OPNSCOPE

Specifies the extent of influence (scope) of the open operation.

***ACTGRPDFN:** The scope of the open operation is determined by the activation group of the program that called the OVRDKTF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

***JOB:** The scope of the open operation is the job in which the open operation occurs.

Examples for OVRDKTF

Example 1: Changing Spooling Specifications

```
OVRDKTF FILE(OUT) VOL(DPT706) LABEL(STATUSR)
        SPOOL(*YES)
```

This command changes the spooling specification for the output file named OUT. When a program produces output data for the OUT file, the data is spooled for processing by a spooling writer. The writer processes the data by writing it in a data file called STATUSR that is on a diskette whose volume identifier is DPT706.

Example 2: Specifying DBCS Processing

```
OVRDKTF FILE(IGCLIB/IGCDCT) IGCDTA(*YES)
```

This command overrides the diskette file IGCDCT, which is stored in the library IGCLIB, so that the file contains double-byte character set data.

Error messages for OVRDKTF

*ESCAPE Messages

CPF180C

Function &1 not allowed.

CPF1892

Function &1 not allowed.


OVRDSPF (Override with Display File) Command Description

OVRDSPF Command syntax diagram

Purpose

The Override with Display File (OVRDSPF) command is used to (1) override (replace) the file named in the program, (2) override certain parameters of a file used by the program, or (3) override the file named in the program and override certain parameters of the file processed. Parameters overridden by this command are specified in the file description, in the program, or in other called file override commands.

If a file named in the program is overridden, the name of that file is specified in the FILE parameter and the name of the overriding file (the file being processed) is specified in the TOFILE parameter. The OVRDSPF command also specifies parameters to override values contained in the file description of the overriding file. If the file named in the program is not replaced but certain parameters of the file are overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters overridden are then specified by the other parameters of the OVRDSPF command. Parameters that are not specified do not affect parameters specified in the file description, in the program, or in other called file override commands.

More information on overriding files is in the File Management topic in the Information Center, and the Application Display Programming  book.

Required Parameter

FILE Specifies the name of the file being used by the program to which this override command is applied. If TOFILE(*FILE) is specified, a display device file must be specified. Otherwise, any device file or database file can be specified.

Optional Parameters

TOFILE

Specifies the qualified name of the display file used instead of the file specified in the FILE

parameter or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified on this OVRDSPF command override the same parameters specified in the display device file, in the program, or in other called OVRDSPF commands.

***FILE:** The display device file named in the FILE parameter has some of its parameters overridden by the values specified in this command.

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.

display-device-file-name: Specify the name of the display device file used instead of the overridden file.

DEV Specifies the names of one or more display devices used with this display device file to pass data records between the users of the display devices and their jobs. The device name specified in the display device file supplied by IBM is *REQUESTER.

This parameter overrides the device names specified in the device file, in the program, or in other called OVRDSPF commands.

***REQUESTER:** The device from which the program is called is assigned to the file when the file is opened.

device-name: Specify the names of one or more display devices used with this device file to pass data records between the users of the devices and the system. Each device name must already be known on the system by a device description before this device file is created. *REQUESTER can be specified as one of the names. Up to 50 names can be specified in this command, but the total number cannot exceed the number specified on the MAXDEV parameter.

CHRID

Specifies the character identifier (graphic character set and code page) that a work station display device supports. When a display file that was created with the CHRID DDS keyword is used with the device, the system converts data sent to and received from the device to ensure that the correct characters are shown and that the correct hexadecimal byte values are returned to the application program. More information about display file CHRID processing and the translation tables that are used to convert data sent to and received from the display are in the Application

Display Programming  book.

***DEV D:** The CHRID value specified in the device description of the work station on which the application is running is used. If no CHRID value is specified, the QCHRID system value (for the system on which the application is running) is used. No translation is necessary because the file has the same character identifier as the work station. For a list of valid values, see the CHRID parameter of the Create Device Description Display (CRTDEV D) command description.

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

***JOBCCSID:** The character data is converted, if necessary, from the device CHRID to the CCSID (coded character set identifier) of the job during input, and from the CCSID of the job to the device CHRID on output.

Note:

This value is not allowed if the file was created on a system at an earlier release level than V2R3M0.

***CHRIDCTL:** The system checks the CHRIDCTL job definition attribute to determine whether to use *JOBCCSID or *DEVD on the CHRID command parameter for this file.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values that match the attributes of the display device. Valid values range from 1 through 32767.

Element 2: Code Page

code-page: Specify the code page set values that match the attributes of the display device. Valid values range from 1 through 32767.

DECFMT

Specifies which decimal format value is used when editing numeric fields with the EDTCDE DDS keyword. The decimal format value determines the use of commas and periods for the decimal position and three digit positional separators on edited fields.

***FILE:** Use the decimal format value stored with the file when the file was created.

***JOB:** Use the decimal format value from the DECFMT job attribute when the file is opened.

SFLENDTXT

Specifies where the 'More...' and 'Bottom' text is retrieved from when displaying a subfile. The 'More...' and 'Bottom' text is displayed in a subfile when the SFLEND(*MORE) DDS keyword is specified on the subfile control record.

***FILE:** Use the 'More...' and 'Bottom' text that is stored in the file during file creation. This text was retrieved from messages CPX6AB1 and CPX6AB2 which exist in the active language of the system when the file was created.

***MSG:** Use the 'More...' and 'Bottom' text retrieved from messages CPX6AB1 and CPX6AB2 which exist in the current active language of the system when the file is opened.

RTNDATCAK

Specifies whether AID keys which do not return data, like CA keys, the print, help, home, and clear keys, will allow input data to be returned from the device to the application after validity checking has caused the input buffer to be updated.

***NO:** The input buffer will be restored to its original values before it is returned to the application. Any date, time or timestamp field which has invalid data is replaced in the input buffer with a valid default value.

***YES:** The input buffer, which may include values that did not pass the validity checks, will be returned to the application. Any date, time, or timestamp field which has invalid data is replaced in the input buffer with a valid default value.

IGCDTA

Specifies, for program-described original files, whether the file processes double-byte character set (DBCS) data. For externally described printer files, this parameter specifies DBCS attributes of the file.

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

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

IGCEXNCHR

Specifies whether the system processes double-byte character set (DBCS) extension characters.

***YES:** The system processes DBCS extension characters.

***NO:** The system does not process DBCS extension characters; it displays extension characters as the undefined character.

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.

This parameter overrides the wait time specified in the device file, in the program, or in other called OVRDSPF commands.

***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 display device file 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.

WAITRCD

Specifies the number of seconds the program waits for the completion of a read-from-invited-device operation to a multiple device file in a high-level language program. Refer to the appropriate high-level language reference manual to determine when a file is treated as a multiple device file. The program performing the read operation waits for input from all invited devices currently accessing the file. If a record is not returned from an invited device in the specified amount of time, a notify message is sent to the program. This parameter has no effect on an input operation directed to a specific device.

Note:

This parameter is also used to specify the time (seconds) that a CL program waits to complete a WAIT command. If a record is not returned from any of the devices that should return a record, an escape message is sent to the CL program. More information on the WAITRCD parameter is in the Receive File (RCVF), Send File (SNDF), Send/Receive File (SNDRCVF), and WAIT (Wait) command descriptions.

This parameter overrides the wait record value specified in the device file, in the program, or in other called OVRDSPF commands.

***NOMAX:** There is no limit on the time the system waits for the completion of the operation.

***IMMED:** The program does not wait for the read-from-invited-device operation for the completion of the file. A record must be available from an invited program device when the

read-from-invited-program-device operation is performed. If a record is not already available when the read-from-invited-program-device operation is performed, a notify message is sent to the program.

number-of-seconds: Specify the number of seconds that the program waits for the completion of the read-from-invited-device operation. Valid values range from 1 through 32767.

LVLCHK

Specifies whether the record format level identifiers in the program are checked against those in the device file when the file is opened. If so, the record format identifiers in the program must match those in the device file. Because the same record format name can exist in more than one file, each record format is given an internal system identifier when it is created.

Note:

This parameter overrides the value specified in the device file, in the program, or in other called OVRDSPF commands. Level checking cannot be done unless the program contains the record format identifiers. This command cannot override level checking from *NO to *YES.

***NO:** The level identifiers are not checked when the file is opened.

SECURE

Specifies whether this file is safe from the effects of previously called file override commands. If SECURE is not specified, processing occurs as if SECURE(*NO) is specified.

***NO:** This file is not protected from the effects of other file overrides; its values can be overridden by the effects of previously called file override commands.

***YES:** This file is protected from the effects of any file override commands previously called.

OVRSCOPE

Specifies the extent of influence (scope) of the override.

***ACTGRPDFN:** The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL:** The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB:** The scope of the override is the job in which the override occurs.

DTAQ

Specifies the name of the data queue that receives an entry from the system when a data-available event is signaled from an invited display device. The data queue need not exist when the display file is created since the name specified on this parameter is not evaluated until

the file is used. More information on the data queue function is in the CL Programming  book.

***NONE:** A data queue does not receive an entry from the system.

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.

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

data-queue-name: Specify the name of the data queue that is to receive an entry from the system when the data-available event is signaled.

SHARE

Specifies whether the open data path (ODP) for the display 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.

This parameter overrides the value specified in the device file, in the program, or in other called OVRDSPF commands.

***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.

OPNSCOPE

Specifies the extent of influence (scope) of the open operation.

***ACTGRPDFN:** The scope of the open operation is determined by the activation group of the program that called the OVRDSPF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

***JOB:** The scope of the open operation is the job in which the open operation occurs.

Example for OVRDSPF

```
OVRDSPF FILE(DISPLAY75) WAITFILE(30)
```

This command overrides the file wait time value specified in the DISPLAY75 device file description, in the program, or in other called OVRDSPF commands. The program in which this command occurs waits up to 30 seconds (if necessary) to allocate the required file resources to the file named DISPLAY75.

Error messages for OVRDSPF

*ESCAPE Messages

CPF1892

Function &1 not allowed.

OVRICFF (Override with Intersystem Communications Function File)

Command Description


OVRICFF Command syntax diagram

Purpose

The Override with Intersystem Communications Function File (OVRICFF) command overrides the file named in the program and overrides certain parameters of the file being processed. Parameters overridden by this command can be specified in the file description, in the program, or in other file override commands that are run later.

If a file named in the program is being overridden, the name of that file is specified in the FILE parameter and the name of the overriding file (the file being processed) is specified in the TOFILE parameter.

This command can also specify parameters to override values contained in the file description of the overriding file. If the file named in the program is not being replaced but certain parameters of the file are being overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters being overridden are then specified by the other parameters of the OVRICFF command. Parameters that are not specified do not affect parameters specified in the file description, in the program, or in other override commands run later.

More information on overriding files is in the File Management topic in the Information Center and the ICF Programming  book.

Required Parameter

FILE Specifies the name of the file being used by the program to which this override command is applied. If TOFILE(*FILE) is specified, a database file must be specified. Otherwise, any device file or database file can be specified.

Optional Parameters

TOFILE

Specifies the qualified name of the ICF file (up to 10 characters in length) that is used instead of the file specified in the FILE parameter or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified in this command override the other values specified in the ICF file or in the program.

***FILE:** Some of the parameters of the ICF file named in the FILE parameter are overridden by values specified in this parameter.

The name of the ICF file 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.

file-name: Specify the qualified name of the physical file or device file that receives the copied records. If no library qualifier is specified, *LIBL is used to locate the file. However, if

CRTFILE(*YES) is specified and the specified file cannot be found, the file name must be qualified with a library name. When the physical to-file is created, it is placed in the specified library.

ACQPGMDEV

Specifies which program device is acquired to the file when the file is opened. This parameter overrides the value in the ICF file, in the program, or in the other OVRICFF commands run later.

***NONE:** The file is opened without any devices acquired. All devices used with this file must be explicitly acquired before input/output is directed to them.

program-device-name: Specify the name of the program device that is acquired when the file is opened. The name should be specified on the Add Intersystem Communications Function Program Device Entry (ADDICFDEVE) command or the Override Intersystem Communications Function Program Device Entry (OVRICFDEVE) command as a program-device-name before the file is opened.

MAXRCLEN

Specifies the maximum record length used when the file is opened. This parameter overrides the value in the ICF file, in the program, or in the other OVRICFF commands run later.

***CALC:** The value calculated in the file is used when the file is opened.

record-length: Specify the maximum record length (in bytes) used when the file is opened. Valid values range from 1 through 32767.

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.

This parameter overrides the wait time specified in the device file, in the program, or in other OVRICFF commands run later.

***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 ICF file 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.

WAITRCD

Specifies the number of seconds the program waits for the completion of a read-from-invited-device operation to a multiple device file in a high-level language program. Refer to the appropriate high-level language reference manual to determine when a file is treated as a multiple device file. The program performing the read operation waits for input from all invited devices currently accessing the file. If a record is not returned from an invited device in the specified amount of time, a notify message is sent to the program. This parameter has no effect on an input operation directed to a specific device.

Note:

This parameter is also used to specify the time (seconds) that a CL program waits to complete a WAIT command. If a record is not returned from any of the devices that should return a record, an escape message is sent to the CL program. More information on the WAITRCD parameter is in the Receive File (RCVF), Send File (SNDF), Send/Receive File (SNDRCVF), and WAIT (Wait) command descriptions.

This parameter overrides the wait record value specified in the device file, in the program, or in other override commands started later.

***NOMAX:** There is no limit on the time the system waits for the completion of the operation.

***IMMED:** The program does not wait for the read-from-invited-device operation for the completion of the file. A record must be available from an invited program device when the read-from-invited-program-device operation is performed. If a record is not already available when the read-from-invited-program-device operation is performed, a notify message is sent to the program.

number-of-seconds: Specify the number of seconds that the program waits for the completion of the read-from-invited-device operation. Valid values range from 1 through 32767.

LVLCHK

Specifies whether the record format level identifiers in the program are checked against those in the device file when the file is opened. If so, the record format identifiers in the program must match those in the device file. Because the same record format name can exist in more than one file, each record format is given an internal system identifier when it is created.

Note:

This parameter overrides the value specified in the device file, in the program, or in other override commands run later. Level checking cannot be done unless the program contains the record format identifiers. This command cannot override level checking from *NO to *YES.

***NO:** The level identifiers of the record formats are not checked when the file is opened.

SECURE

Specifies whether this file is safe from the effects of previously called file override commands. If SECURE is not specified, processing occurs as if SECURE(*NO) is specified.

***NO:** This file is not protected from the effects of other file overrides; its values can be overridden by the effects of previously called file override commands.

***YES:** This file is protected from the effects of any file override commands previously called.


OVRSCOPE

Specifies the extent of influence (scope) of the override.

***ACTGRPDFN:** The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL:** The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB:** The scope of the override is the job in which the override occurs.

DTAQ Specifies the name of the data queue that receives an entry from the system when a data-available event is signaled from an invited display device. The data queue need not exist when the display file is created since the name specified on this parameter is not evaluated until the file is used. More information on the data queue function is in the CL Programming  book.

***NONE:** A data queue does not receive an entry from the system.

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.

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

data-queue-name: Specify the name of the data queue that is to receive an entry from the system when the data-available event is signaled.

SHARE

Specifies whether the open data path (ODP) for the ICF 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.

OPNSCOPE

Specifies the extent of influence (scope) of the open operation.

***ACTGRPDFN:** The scope of the open operation is determined by the activation group of the program that called the OVRICFF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

***JOB:** The scope of the open operation is the job in which the open operation occurs.

Example for OVRICFF

```
OVRICFF FILE(ICFHIST) TOFILE(PRSNNL/ICFCURT)
```

This command overrides the file named ICFHIST to the ICF file named ICFCURT in library PRSNL.

Error messages for OVRICFF

*ESCAPE Messages

CPF1892

Function &1 not allowed.

OVRICFDEVE (Override with Intersystem Communications Function Program Device Entry) Command Description

OVRICFDEVE Command syntax diagram

Purpose

The Override with Intersystem Communications Function Program Device Entry (OVRICFDEVE) command can be used to temporarily add the program device entry and the remote location name to the ICF file or to override a program device entry with the specified remote location name and attributes for an ICF file.

More information on how overrides processing is performed is in the File Management topic in the

Information Center, the Application Display Programming  book, and the Printer Device Programming



book.

Required Parameter

PGMDEV

Specifies the program device name for an ICF file whose attributes are being overridden. The total number of devices that may be added to an ICF file is determined by the MAXPGMDEV parameter on the Create ICF File (CRTICFF) command or the Change ICF File (CHGICFF) command.

Specify the name of the ICF program device entry with which the program communicates. This name is used in device-specific input/output operations to identify the program device and the session attributes. The program device name must be unique, although the same remote location name may be specified more than once. This allows more than one session to be at the same remote location, and/or to have different attribute values for each session at the same remote location. This program device name must be unique throughout the entries for the ICF file. If an override command is entered a second time for the same program device, then both (according to the override process rules) define the same program device entry.

Optional Parameters

Note:

Refer to the APPC, APPN, and HPR topic in the Information Center for information on how the system uses the RMTLOCNAME, DEV, LCLLOCNAME, and RMTNETID parameters to select an APPC device description.

RMTLOCNAME

Specifies the remote location name of the system with which this object communicates.

Note:

A remote location must be specified by using the Add Intersystem Communications Function Program Device Entry (ADDICFDEVE) command or an applied program device override. If a remote location is not specified, an escape message is sent when the program device is acquired.

***REQUESTER:** The name used to refer to the communications device through which the program is started is used. The session that is assigned when the program device is acquired is the same session in which the program start request is received. If the program is not started as a result of a program start request, the acquire operation of the program device fails. The target program uses *REQUESTER as the remote location name in the intersystem communications function (ICF) file to connect to the session that the source program used to send the program start request.

The *REQUESTER value can be specified on only one program device entry and is valid only for a target communication job. If *REQUESTER is specified in any other type of job, a message is sent.

remote-location-name: Specify the full name of a remote location. The remote location does not need to exist when this command is run, but it must exist (be configured on the system or in the advanced peer-to-peer networking (APPN) support for this remote location) at the time the program acquires the program device. A given remote location may be added many times by using different program device names. When a program is running, however, only one program device name associated with each asynchronous or binary synchronous communications equivalence link (BSCCEL) or system network architecture upline facility (SNUF) remote location may read or change the file at any one time. For each remote advanced program-to-program communications (APPC), INTRA, or SNUF location, more than one associated program device name may be acquired to the file at one time. Each SNUF remote location may have many devices. The system determines which device to use unless a device is specified on the DEV parameter.

CMNTYPE

Specifies which communications types are used. This parameter is used only for prompting purposes; it is ignored when the command is run. The value specified for this parameter determines the subset of other parameters that are displayed (prompted) for the user.

***ALL:** The parameters for all of the communications types appear in the prompt.

***APPC:** The advanced program-to-program communications (APPC) parameters appear in the prompt.

***ASYNC:** The asynchronous (ASYNC) parameters appear in the prompt.

***BSCCEL:** The binary synchronous communications equivalence link (BSCCEL) parameters appear in the prompt.

***FINANCE:** The finance parameters appear in the prompt.

***INTRA:** The intrasystem (INTRA) parameters appear in the prompt.

***RETAIL:** The retail parameters appear in the prompt.

***SNUF:** The SNA upline facility (SNUF) parameters appear in the prompt.

DEV

Specifies the communications device used in the remote location. This parameter should be specified only for APPC, INTRA, and SNUF communications types. If the Add Intersystem Communications Function Program Device Entry (ADDICFDEVE) command is not run for the specified program device and this parameter is not overridden, DEV(*LOC) is used.

***LOC:** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

device-name: Specify the name of a communications device associated with the remote location. If the device name is not valid for the remote location, a message is sent when the program device entry is acquired. More information on device names is in the APPC, APPN, and HPR topic in the Information Center.

LCLLOCNAME

Specifies the local location name.

Note:

This parameter applies to the APPC communications type only and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device or is not overridden, this parameter defaults to *LOC.

***LOC:** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

***NETATR:** The LCLLOCNAME value specified in the system network attributes is used.

local-location-name: Specify the name of the user's location. If the local location name is not valid for the remote location or remote location and device, an escape message is sent when the program device entry is acquired.

MODE Specifies the mode name used. This parameter applies only to the APPC communications type and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, MODE(*NETATR) is used.

***NETATR:** The mode name specified in the network attributes is used.

***BLANK:** The mode name consisting of 8 blank characters is used.

mode-name: Specify a mode name for the APPC communications device. If the specified mode is not valid for any combination of remote location, device, local location, and remote network ID, an escape message is sent when the program device entry is acquired.

RMTNETID

Specifies the remote network ID used with the remote location. This parameter applies to the APPC communications type only and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, RMTNETID(*LOC) is used.

***LOC:** The remote network identifier (ID) associated with the remote location is used. If several remote network IDs are associated with the remote location, the system determines which remote network ID is used.

***NETATR:** The RMTNETID value specified in the system network attributes is used.

***NONE:** No remote network identifier (ID) is used.

remote-network-ID: Specify a remote network ID for the APPC communications device.

FMTSLT

Specifies the record format selection used for input operations. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, FMTSLT(*PGM) is used.

***PGM:** The program determines which record formats are selected. If an input (read) operation with a record format name is specified, that format is selected. If an input operation without a

record format is specified, the default format (the first record format in the file) is selected. This also means that if there are any record identification (RECID) parameters specified in the data description specifications (DDS) for the file, or if any remote formats are received, they are not taken into consideration when the record is selected.

***RECID:** The record identification (RECID) keywords specified in the DDS for the file are used to do record selection. If there are no RECID keywords in the file, an error message is sent, the acquire operation of the program device ends, and the device is not acquired.

***RMTFMT:** The remote format names received from the sending system are used to do record selection. If the device is not an APPC or intrasystem device and *RMTFMT is specified, a run time error occurs at the time the program device entry is acquired.

APPID

Specifies (in characters) the VTAM identifier of the Customer Information Control System for Virtual Storage (CICS/VS) or Information Management System/Virtual Storage (IMS/VS) host subsystem sent with the sign-on message. This parameter applies to the SNUF communications type only and is ignored for all other devices. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, APPID(*DEV) is used.

***DEV:** The application identifier specified in the device description is sent with the sign-on message.

***USER:** The application program can send messages or a logon to the host. This is valid only when using the 3270 program interface.

application-ID: Specify the application identifier that is sent with the sign-on message.

BATCH

Specifies, for CICS/VS and IMS/VS, whether this session is used for batch jobs. This parameter applies to the SNUF, Retail, and INTRA communications types only and is ignored for all other devices. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, BATCH(*NO) is used.

***NO:** Batch jobs do not occur.

***YES:** Batch jobs occur and SNUF is not to assemble physical records into logical records. If BATCH(*YES) is specified, MSGPTC(*NO) must also be specified.

HOST Specifies the host or remote subsystem with which this session communicates. This parameter applies to the SNUF communications type only and is ignored for all other devices. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, HOST(*DEV) is used.

***DEV:** The host system specified in the device description is used.

***CICS:** The session communicates with CICS/VS.

***IMS:** The session communicates with IMS/VS.

***IMSRTR:** The session communicates with IMS/VS using the ready-to-receive option.

ENDSSNHOST

Specifies how SNUF ends the session with the host.

Note:

This parameter applies to SNUF communications type only and is ignored for all other devices.

***RSHUTD:** SNUF sends a request-shut-down command to the host.

***TERMSELF:** SNUF sends a terminate-self command to the host. It may be necessary to specify this value if the value *RSHUTD fails to end a session with a non-IBM host.

SPCHOSTAPP

Specifies whether SNUF customizes support for special host applications outside the CICS or IMS application layer.

***DEV D:** The special host application specified in the device description is used.

***NONE:** SNUF does not customize support for special host applications.

***FLASH:** SNUF customizes support for the Federal Link Access for Secondary Half-sessions (*FLASH) protocol application.

INZSELF

Specifies whether a formatted INIT-SELF is built in place of the unformatted sign-on normally sent by SNUF to the host.

***NO:** The unformatted default sign-on provided by SNUF is used.

***YES:** The formatted INIT-SELF provided by SNUF is used.

HDRPROC

Specifies, for both CICS/VS and IMS/VS, whether received function management headers are passed to the application program. This parameter applies to the SNUF communications type only and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, HDRPROC(*SYS) is used.

***SYS:** SNA upline facility (SNUF) removes function management headers before passing data to the program.

***USER:** Function management headers are passed with the data to the program.

MSGPTC

Specifies, for both CICS/VS and IMS/VS, whether message protection is used for this session. This parameter applies to the SNUF communications type only and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device, MSGPTC(*YES) is used.

***YES:** Message protection is used. SNUF saves messages until they are responded to and tries resynchronization if errors occur. *YES is only valid when BATCH(*NO) is also specified.

***NO:** Message protection is not used.

EMLDEV

Specifies that the program device entry is used to send and receive data streams to and from specific types of 3270 display or printer devices being emulated. This parameter consists of an emulation device type and an emulation device data format. The emulation device data format specifies the format of the type 3270 data stream being sent or received. A 20- or 32-byte common header that contains type 3270 command and data flow information is located at the start of the I/O buffer that is sending or receiving the type 3270 data stream. This parameter applies to the SNUF communications type only. This parameter can be specified as a list of two values (elements) or as a single value (*NONE).

***NONE:** This program device entry is not used for sending and receiving 3270 data streams.

Element 1: Type of Device

3278: The data stream is for a 3279, 3278 or 3277 display device.

3284: The data stream is for a 3284 Printer.

3286: The data stream is for a 3286 Printer.

3287: The data stream is for a 3287 Printer.

3288: The data stream is for a 3288 Printer.

3289: The data stream is for a 3289 Printer.

Element 2: Format of Data Stream

***UNFORMAT:** An unformatted 3270 data stream is sent or received. The user application program must translate the data stream into a display or printer image.

***FIELD:** A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display or printer image followed by field definitions. The field definitions indicate the location and characteristics of each field.

***NOFIELD:** A formatted 3270 data stream that has no field definitions but contains a display or printer image is sent or received.

Note:

If *FIELD or *NOFIELD is specified, BATCH(*NO) must also be specified.

***EXTFIELD:** A formatted 3270 data stream contains extended field attribute information. The extended field attribute information is in the field definitions which follow the display image. The field definitions indicate the location and characteristics of each field. This value can only be specified if 3278 is also specified for the type of device on the EMLDEV parameter.

CNVTYPE

Specifies the conversation type for which the application program is designed. This parameter is valid only for the APPC communications type and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, CNVTYPE(*SYS) is used.

More information on the APPC communications type is in the APPC, APPN, and HPR topic in the Information Center.

***SYS:** The advanced program-to-program communications (APPC) mapped conversation support is used.

***USER:** The advanced program-to-program communications (APPC) basic conversation support is used.

***SRCPGM:** The target program accepts the conversation type specified by the source program.

BLOCK

Specifies whether the system or the user controls the combination of records into blocks when they are sent. This parameter is for the BSCEL communications type only and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, BLOCK(*DEVVD) is used.

With this parameter, specify one of the following conditions of record blocking:

- No blocking/deblocking: The record format described in the DDS is the format for both the record and the block.
- User blocking and deblocking: Give the BSC controls needed to describe the record format of the system.
- System blocking with record separator characters: Specify the record separator character used by the system to determine record boundaries in the block.
- System blocking of fixed-length records: The system uses fixed-length records, and blocks and deblocks records accordingly.

If a parameter value other than *NONE or *USER is specified, records are blocked as required by the system for output and are deblocked on input.

Element 1: Blocking Option

***DEV D:** The block option in the device description is used.

***NONE:** Blocking or deblocking is not done by the system.

***ITB:** The records are blocked or deblocked, based on the location of an Intermediate Text Block (ITB) control character. For input files, a record is delimited by locating the next intermediate text block character. An end-of-text or end-of-transmission block character is used as an intermediate text block character to delimit a block. For output files, an ITB character is added after the record. If it is the last character of the block, the ITB is replaced by an end-of-text or end-of-transmission block character.

***IRS:** The records are blocked or deblocked based on the location of an Interrecord Separator (IRS) character. For input files, a record is delimited by locating the next IRS character. For output files, an IRS character is added after the record.

***NOSEP:** No record separator character is contained in the block that is sent to or received from the device. The system blocks and deblocks the records using a fixed-length record, as specified in the DDS format specifications.

***USER:** The program provides all the control characters (including record separator characters, binary synchronous communications (BSC) framing characters, and transparency characters) necessary to send records. More information about the device and binary synchronous communications equivalence link (BSC) support characteristics is in the BSC Equivalence Link

Programming  book.

***SEP:** Records are blocked or deblocked, based on the location of a user-specified record separator character. For input files, a record is delimited by locating the next record separator character. For output files, a record separator character is added after the record.

Element 2: Record Separator

X'1E': The record separator X'1E' is used.

record-separator-character: Specify a unique 1-byte record separator character. The record separator character may be specified as 2 hexadecimal characters, as in BLOCK(*SEP X'FD'), or as a single character, as in BLOCK(*SEP @). If a record separator character is not specified, the record separator character X'1E' is used.

The following is a list of BSC control characters that should not be used as record separator characters:

EBCDIC	ASCII	BSC Control
X'01'	X'01'	SOH (start-of-header)
X'02'	X'02'	STX (start-of-text)
X'03'	X'03'	ETX (end-of-text)
X'10'	X'10'	DLE (data-link escape)
X'1D'	X'1D'	IGS (interchange group separator)
X'1F'	X'1F'	ITB (intermediate text block)
X'26'	X'17'	ETB (end-of-transmission block)
X'2D'	X'05'	ENQ (enquiry)
X'32'	X'16'	SYN (synchronization)
X'37'	X'04'	EOT (end-of-transmission)
X'3D'	X'15'	NAK (negative acknowledgment)

RCDLEN

Specifies the maximum record length (in bytes) for data sent and received. This parameter applies to the SNUF and the BSCEL communications types only and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, RCDLEN(*DEVD) is used.

***DEVD:** The record length specified in the device description is used. If a record is longer than the specified record length, a run time error occurs at the time the record is sent or received.

record-length: Specify the maximum record length (in bytes) to use with this device file. The value must be at least the size of the largest record sent. If a record is longer than the specified record length, a run time error occurs when the record is sent or received. Valid values range from 1 through 32767 bytes for SNUF communications. For BSCEL communications, the maximum record length is 8192 bytes.

BLKLEN

Specifies the maximum block length (in bytes) for data sent. This parameter applies to the BSCEL and the SNUF communications types and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, this parameter defaults to *DEVD.

***DEVD:** The block length specified in the device description is used.

block-length: Specify the maximum block length (in bytes) of records sent. The value must be at least the size of the largest record sent. Valid values range from 1 through 32767 for SNA upline facility (SNUF). For binary synchronous communications equivalence link (BSCEL) communications, the maximum block length is 8192.

TRNSPY

Specifies whether text is sent in transparent text mode. Transparent text mode sends all 256 extended binary-coded decimal interchange code (EBCDIC) character codes; use this feature when sending packed or binary data fields. This parameter is for the BSCEL communications type only and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, TRNSPY(*DEVD) is used.

***DEVD:** The text transparency option specified in the device description is used.

***NO:** Text transparency is not used.

***YES:** Text transparency is used, which sends all 256 EBCDIC character codes. *YES is valid only when BLOCK(*NONE), BLOCK(*NOSEP), or BLOCK(*USER) is specified.

Note:

Transparency of received data is determined by the data stream; therefore, this parameter is not relevant for received data. If TRNSPY(*YES) is specified with BLOCK(*USER), BSCEL ignores the transparency indicator during write operations. Correct controls must be given with the data to get transparent transmission of data. For example, first specify the DLE and STX control characters; the system gives the remaining control characters for transparent transmission of data.

DTACPR

Specifies whether data compression is performed.

Note:

This parameter is for the BSCCEL communications type only and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, DTACPR(*DEVD) is used.

***DEVD:** The data compression option specified in the device description is used.

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

***YES:** Data is compressed for output and decompressed for input. DTACPR(*YES) cannot be specified if TRNSPY(*YES) is specified.

TRUNC

Specifies whether trailing blanks are removed from any output records. TRUNC(*YES) cannot be specified if BLOCK(*NOSEP) or BLOCK(*ITB) is specified. If TRUNC(*YES) is specified when DTACPR(*YES) or BLOCK(*USER) is specified, truncation is ignored. This parameter is for BSCCEL communications types only and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, TRUNC(*DEVD) is used.

***DEVD:** The trailing blanks specified in the device description are used.

***NO:** Trailing blanks are not removed from output records.

***YES:** Trailing blanks are removed from output records.

OVRFLWDTA

Specifies whether overflow data is discarded or retained.

***DISCARD:** Overflow data is not kept.

***RETAIN:** Overflow data is kept.

GRPSEP

Specifies a separator for groups of data (for example, data sets and documents). This parameter is for the BSCCEL communications type only and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, GRPSEP(*DEVD) is used.

***DEVD:** The group separator option specified in the device description is used.

***EOT:** The end-of-transmission (EOT) BSC control character is used as a data group separator.

***DEV3740:** A null record (STXETX) is used as a data group separator.

***OFCSYS:** A block sent with the end-of-text (ETX) BSC control character is used as a data group separator.

RMTBSCCEL

Specifies whether the type of BSCCEL session is with a BSCCEL system. This parameter applies to the BSCCEL communications type only and is ignored for all other communications types. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, RMTBSCCEL(*DEVD) is used.

***DEVD:** The RMTBSCCEL option specified in the device description is used.

***NO:** A RMTBSCCEL attribute of *NO indicates that the remote system cannot recognize BSCCEL commands or messages. In most cases, *NO is used when communicating with remote systems such as a 3741 Data Entry Station, an Office System 6, a 5230 Data Collection System, or a System/38.

***YES:** The remote system recognizes the BSCCEL transaction starting commands, transaction ending commands, and online messages. In most cases, *YES indicates that the remote system is another iSeries 400 computer, AS/400 system, System/38, System/36, or a System/34 with BSCCEL support.

INLCNN

Specifies the method used to make a connection on the line for the session being acquired. This parameter applies to the binary synchronous communications equivalence link (BSCCEL) communications types only.

***CTLD:** The initial connection option specified in the controller description is used.

***DIAL:** The local system starts the call.

***ANS:** The remote system starts the call and the local system answers the call.

SECURE

Specifies whether this file is safe from the effects of previously called file override commands. If SECURE is not specified, processing occurs as if SECURE(*NO) is specified.

***NO:** This file is not protected from the effects of other file overrides; its values can be overridden by the effects of previously called file override commands.

***YES:** This file is protected from the effects of any file override commands previously called.

OVRSCOPE

Specifies the extent of influence (scope) of the override.

***ACTGRPDFN:** The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL:** The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB:** The scope of the override is the job in which the override occurs.

Examples for OVRICFDEVE

Example 1: Overriding the Device Entry with the Record Format Selection Attributes

```
OVRICFDEVE PGMDEV(BSCCEL2) RMTLOCNAME(BSCNYC)
           FMTSLT(*RECID)
```

This command overrides the program device named BSCCEL2 with a corresponding remote location named BSCNYC for any ICF file associated with the job. The program device is overridden with the attributes of FMTSLT(*RECID).

Example 2: Overriding the Device Entry with the Record Format Selection and the Conversation Type Attributes

```
OVRICFDEVE PGMDEV(APPC1) RMTLOCNAME(*REQUESTER)
           FMTSLT(*RMTFMT) CNVTYPE(*SYS)
```

This command overrides the program device entry named APPC1 with a remote location name of *REQUESTER. The program device entry is overridden with the FMTSLT(*RMTFMT) and CNVTYPE(*SYS) attributes.

Example 3: Overriding an Entry for Associated ICF Files

```
OVRICFDEVE PGMDEV(JOE) RMTLOCNAME(LU0MPLS)
```

This command overrides the program device entry named JOE with a remote location named LU0MPLS for any ICF file associated with the job.

Example 4: Specifying the Communications Device

```
OVRICFDEVE PGMDEV (APPC) RMTLOCNAME (APPCMPLS)
DEV (MPLSLINE2)
```

This command overrides the program device entry named APPC with a remote location named APPCMPLS using device MPLSLINE2.

Error messages for OVRICFDEVE

*ESCAPE Messages

CPF180C

Function &1 not allowed.

CPF1892

Function &1 not allowed.

OVRMSGF (Override with Message File) Command Description

OVRMSGF Command syntax diagram

Purpose

The Override with Message File (OVRMSGF) command overrides a message file used in a program. The overriding message file (specified in the TOMSGF parameter) is used whenever a message is sent or retrieved and the overridden message file is specified.

The overriding message file need not contain all the messages that the overridden file contains. When a message is received or retrieved and the message identifier cannot be found in the overriding message file, the overridden message file is searched for the identifier. Overriding message files can be overridden, resulting in a chain of overrides. This chain of overrides provides a list of message files that are searched when a message is received or retrieved. Up to 30 message file overrides can be specified in a program.

Restrictions:

1. In a multithreaded job, this command may only be issued from the initial thread.
2. In a multithreaded job, this command will only affect message file references performed in the initial thread. Message file references performed in secondary threads will be unaffected.

More information on overriding files is in the File Management topic in the Information Center.

Required Parameters

MSGF Specifies the name of the message file being used by the program to which this override command is applied.

TOMSGF

Specifies the qualified name of the message file that is used instead of the message file specified in the MSGF parameter or, if the names are the same, specifies that the SECURE parameter specified in the command is used for the message file.

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

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

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

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

message-file-name: Specify the name of the message file to use.

Optional Parameter

SECURE

Specifies whether this file is safe from the effects of previously called file override commands. If SECURE is not specified, processing occurs as if SECURE(*NO) is specified.

***NO:** This file is not protected from the effects of other file overrides; its values can be overridden by the effects of previously called file override commands.

***YES:** This file is protected from the effects of any file override commands previously called.

Example for OVRMSGF

```
OVRMSGF  MSGF(WSUSRMSG)  TOMSGF(ORDENTMSGD)
```

This override command causes the defaults for messages stored in ORDENTMSGD to be used instead of defaults stored in WSUSRMSG (which contains messages designed for work station users). As a result of this command, the messages received by the order entry users are tailored to their own environment.

Error messages for OVRMSGF

*ESCAPE Messages

CPF180C

Function &1 not allowed.

OVRPRTF (Override with Printer File) Command Description

OVRPRTF Command syntax diagram


Purpose

The Override with Printer File (OVRPRTF) command is used to (1) override (replace) the file named in the program, (2) override certain parameters of a file that are used by the program, or (3) override the file named in the program and override certain parameters of the file processed. Parameters overridden by this command are specified in the file description, in the program, or in other file override commands run in the following command.

If a file named in the program is overridden, the name of that file is specified in the FILE parameter and the name of the overriding file (the file processed) is specified in the TOFILE parameter. The OVRPRTF command also specifies parameters to override values contained in the file description of the overriding file. If the file named in the program is not replaced but certain parameters of the file are overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters overridden are then specified by the other parameters of the OVRPRTF command. Parameters not specified do not affect parameters specified in the file description, in the program, or in other file override commands run later.

Restrictions:

1. In a multithreaded job, this command may only be issued from the initial thread.
2. In a multithreaded job, only overrides scoped to the job or an ILE activation group will affect opens performed in a secondary thread.

More information on overriding files is in the File Management topic in the Information Center and the Printer Device Programming  book.

Required Parameter

FILE Specifies the name of the file being used by the program to which this override command is applied. If TOFILE(*FILE) is specified, a display device file must be specified. Otherwise, any device file or database file can be specified.

***PRTF:** The *PRTF file override is applied. This override applies to all printer files being opened except for those printer files that already have specific overrides. For example, if a *PRTF override is issued at call level 3, and an override is issued for QSYSPRT at call level 3, the *PRTF override is applied to all printer files being opened except for QSYSPRT since there is a specific override for it.

file-override-name: Specify the names of one or more overridden files for which the overrides in the call level are applied.

Optional Parameters

TOFILE

Specifies the qualified name of the printer file that is used instead of the file specified in the FILE parameter. If *FILE is specified, this parameter specifies that certain attributes are overridden by the parameters specified in this command. The parameters specified on this OVRPRTF command override the same parameters specified in the printer file, in the program, or in other called OVRPRTF commands.

***FILE:** The printer file named in the FILE parameter has some of its parameters overridden by values specified in this command.

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.

printer-device-file-name: Specify the name of the printer file that is used instead of the overridden file.

DEV Specifies the name of a printer device description. For nonspooled output, this identifies the printer device used to produce the printed output. For spooled output, the file is placed on the output queue determined by the OUTQ parameter. If OUTQ(*DEV) is used, the file is placed on the output queue with the same name as the device.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

***JOB:** The printer device specified in the job description is used.


device-name: Specify the name of the printer associated with this display station. The printer and the display station must be attached to the same controller. When printing double-byte character set (DBCS) data, specify a DBCS printer (5553 or 5583).

DEVTYPE

Specifies the type of data stream created for a printer file.

***SCS**: An SNA character stream (SCS) is created. This parameter must be specified when using the 3287, 3812 SCS, 3816 SCS, 4214, 4234 SCS, 4245, 5219, 5224, 5225, 5256, 5262, 6252, or 6262 work station printers.

- If *SCS is specified and the spooled printer file is directed to an IPDS* printer, the SCS printer file is converted to emulate an IPDS printer file. More information is in the Printer Device

Programming  book.

Double-Byte Character Set Consideration:

When using the 5553 and 5583 DBCS-capable printers, DEVTYPE(*SCS) must be specified.

***IPDS**: An intelligent printer data stream* (IPDS*) is created. This parameter can be specified when using an IPDS printer.

- If *IPDS is specified and the spooled printer file is directed to a printer other than an IPDS printer, the IPDS printer file is converted to an SCS printer file. More information is in the Printer

Device Programming  book.

***USERASCII**: An ASCII data stream is placed on a spooled output queue. The user is responsible for placing the entire hexadecimal data stream in the buffer, since the iSeries 400 does not change or validate the values that are passed. This value cannot be specified with SPOOL(*NO).

***AFPDS**: An advanced function print data stream (AFPDS) is created. Some systems refer to this data stream as MODCA-P. *AFPDS spooled files require PSF/400 to print on an IPDS attached printer or Host Print Transform to print on an ASCII attached printer.

***AFPDSL**: Mixed data (line data and AFPDS data) is created. This value can be specified when using the 3812 IPDS, 3816 IPDS, 3820, 3825, 3827, 3828, 3829, 3831, 3835, 3900, 3912, 3916, 3930, 3925, 4028, 4224, 4230, 4234, 4312, 4317, 4324, 6406, 6408, or 6412 IPDS printers. Also for the InfoPrint 20, InfoPrint 32, InfoPrint 40, InfoPrint 60, InfoPrint 3000, and InfoPrint 4000 printers. *AFPDSL spooled files require PSF/400 to print on an IPDS attached printer. The printer must be configured with AFP(*YES).

***LINE**: Line data is created. This value can be specified when using the 3812 IPDS, 3816 IPDS, 3820, 3825, 3827, 3828, 3829, 3831, 3835, 3900, 3912, 3916, 3930, 3925, 4028, 4224, 4230, 4234, 4312, 4317, 4324, 6406, 6408, or 6412 IPDS printers. Also for the InfoPrint 20, InfoPrint 32, InfoPrint 40, InfoPrint 60, InfoPrint 3000, and InfoPrint 4000 printers. *LINE spooled files require PSF/400 to print on an IPDS attached printer. The printer must be configured with AFP(*YES).

PAGESIZE

Specifies the length and width of the printer forms used by this printer file. The length is specified in lines per page or by the units specified for the UOM parameter. The width is specified in print positions (characters) per line or by the units specified for the UOM parameter.

The page size must be specified with reference to the way the data is printed on the page. For example, if using 8.5 inch wide by 11.0 inch long forms and printing at 6 lines per inch with a 10-pitch font, specify PAGESIZE(66 85) PAGRTT(0). However, to rotate the page, specify the page size for an 11.0 inch wide by 8.5 inch long page and enter PAGESIZE(51 110) PAGRTT(90).

Note:


Specify PAGRTT(*AUTO) or PAGRTT(*DEV) and PRTQLTY(*DRAFT) on this command to enable automatic reduction or rotation if the data does not fit on the paper.

Specify PAGRTT(*COR) on this command to enable automatic reduction whether or not the data fits on the paper.

This parameter overrides the form size values specified in the printer file, in the program, or in other called OVRPRTF commands.


Element 1: Page Length Value

page-length: Specify the page length that is used by this printer file. Although a value ranging from 1 through 255 can be specified as the page length, the value specified must not exceed the actual length of the forms used.

More information about the page lengths that are valid for each printer type is in Printer Device Programming  book.

Element 2: Page Width Value

page-width: Specify the page width used by this printer file. The value specified must not exceed the actual width of the forms used.

More information about page width is in the Printer Device Programming  book.

Element 3: Method of Measure


***ROWCOL:** Page length and page width are measured as numbers of rows and columns.

***UOM:** Page length and page width are measured in the units specified on the UOM parameter.

LPI Specifies the line spacing setting on the printer, in lines per inch, used by this printer file.

The line spacing on the 5256 printer must be set manually. When the lines per inch (LPI) value on this parameter changes (from the value on the previous printer file), an inquiry message is sent to the message queue associated with the printer that requests a change to the LPI value.

The line spacing on the 4214, 4224, 4230, 4234, 4245, and 5262 Printers is set by a print command. These also allow setting the lines per inch spacing on the control panel of the printer. The lines per inch value must not be set at the printer. If the LPI value is overridden at the control panel, the system overrides the value set with the LPI value of the next printer file received.

More information about the lines per page and lines per inch that are valid for each printer type is in the Printer Device Programming  book.

This parameter overrides the overflow value specified in the printer file, in the program, or in other called OVRPRTF commands.

3: The line spacing on the printer is 3 lines per inch. This value is valid only for double-byte character set (DBCS) data.

4: 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.

6: The line spacing on the printer is 6 lines per inch.

7.5: The line spacing on the printer is 7.5 lines per inch. This value is valid only for double-byte character set (DBCS) printers.

8: The data density on the tape volume is 38,000 bits per inch, which is used for 1/2 inch reel tapes.


9: The line spacing on the printer is 9 lines per inch.

12: The line spacing on the printer is 12 lines per inch.

CPI Specifies the printer character density, in characters per inch (CPI), used by this printer file.

For the printers that support fonts, the value specified in the font special value implies the CPI. If FONT(*CPI) is specified, the font used is based on the CPI value. The following diagram describes the default font ID for each CPI value:

CPI	FONT ID DEFAULT
5	245
10	011
12	087
13.3	204
15	222
16.7	400
18	252
20	281

More information about the characters per page and characters per inch that are valid for each printer type is in the Printer Device Programming  book.

This parameter overrides the value specified in the printer file, in the program, or in other called OVRPRTF commands.

5: The format of this tape is QIC525, which is used for 1/4 inch cartridge tapes that can hold 525 megabytes of data.

10: Character density is 10 characters per inch.

12: Character density is 12 characters per inch.

13.3: Character density is 13.3 characters per inch. This value is valid only for double-byte character set (DBCS) printers.

15: Character density is 15 characters per inch.

16.7: Character density is 16.7 characters per inch.

18: Character density is 18 characters per inch. This value is valid only on double-byte character set (DBCS) printers.

20: The format of this tape is QIC120, which is used for 1/4 inch cartridge tapes that can hold 120 megabytes of data.

FRONTMGN

Specifies the offset, down and across, of the origin from the edge on the front side of the paper. The offsets are in the units of measure specified on the UOM parameter. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

***DEVD:** The no-print border from the printer is used to place the text on the page when printing to a printer configured as AFP(*YES). A margin of 0 is used for IPDS* printers without a no-print border, or which are configured as AFP(*NO).

Element 1: Offset Down

0: The format of this tape is FMT3570. The data format is written on the tape volume with a 3570 device.

offset-down: Specify the offset of the origin from the top of the page.

Element 2: Offset Across

0: The format of this tape is FMT3570. The data format is written on the tape volume with a 3570 device.

offset-across: Specify the offset of the origin from the left side of the page.

BACKMGN

Specifies the offset, down and across, of the origin from the edge on the back side of the paper. The offsets are in the units of measure specified on the UOM parameter. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

***FRONTMGN:** The offsets specified on the FRONTMGN parameter are used.

***DEVD:** The no-print border from the printer is used to place the text on the page when printing to a printer configured as AFP(*YES). A margin of 0 is used for IPDS* printers without a no-print border, or which are configured as AFP(*NO).

Element 1: Offset Down

0: The format of this tape is FMT3570. The data format is written on the tape volume with a 3570 device.

offset-down: Specify the offset of the origin from the top of the page.


Element 2: Offset Across

0: The format of this tape is FMT3570. The data format is written on the tape volume with a 3570 device.

offset-across: Specify the offset of the origin from the left side of the page.

OVRFLW

Specifies the line number on the current page at which overflow to a new page begins. Generally, after the specified line is printed, the printer overflows to the next page before printing continues. Margins specified for the printer file are ignored when determining overflow. More information is in

the Printer Device Programming  book. This parameter overrides the overflow value specified in the printer file, in the program, or in other called OVRPRTF commands.

overflow-line-number: Specify the line number on the current page at which overflow to a new page begins, whether or not printing has occurred on that line. The value specified must not be greater than the page length (PAGESIZE). Margins specified for the printer file are ignored when determining overflow.

FOLD Specifies whether all positions in a record are printed when the record length exceeds the page

width (specified by the PAGESIZE parameter). When folding is specified and a record exceeds the page width, any portion of the record that cannot be printed on the first line continues (is folded) on the next line or lines until the entire record has been printed.

The FOLD parameter is ignored under the following conditions:

- When DEVTYPE(*SCS) is not specified.
- When printing through OfficeVision*.
- When in the S/36 execution environment.

Double-Byte Character Set Considerations:

The system ignores this parameter when printing double-byte character set (DBCS) files. The system assumes that DBCS records fit on a printed line. If the record exceeds the page width, the system continues printing the record on the next line.

This parameter overrides the value specified in the printer file, in the program, or in other called OVRPRTF commands.

***YES:** Records whose length exceeds the page width are folded on the following lines. Records whose length exceeds the form width are folded on the following lines.

***NO:** Records are not folded; if a record is longer than the page width, only the part of the record that fits on one line is printed.

RPLUNPRT

Specifies (1) whether unprintable characters are replaced and (2) which substitution character (if any) is used. An *unprintable* character is a character the printer is unable to print.

Double-Byte Character Set Considerations:

For double-byte character set (DBCS) data, an unprintable character is one that cannot be processed. When using DBCS-capable printers, consider the following:

- If IGCEXNCHR(*YES) is also specified, the system replaces unprintable extension characters with DBCS underline characters. There may be some cases in which the system is unable to replace an unprintable character with a DBCS underline character. In this case, the undefined character is printed.
- If IGCEXNCHR(*NO) is also specified, the device replaces all extension characters with the undefined character. Choosing a blank as the replacement character for alphanumeric characters might improve system performance.

More information is in the Printer Device Programming  book.

Element 1: Replace Character

***YES:** Unprintable characters are replaced. The program is not notified when unprintable characters are detected. Note the DBCS considerations above.

***NO:** Unprintable characters are not replaced. When an unprintable character is detected, a message is sent to the program.

Element 2: Replacement Character

' ': Specify, if *YES is also specified on this parameter, that a blank is used as the substitution character when an unprintable character is detected.

'replacement-character': Specify, if *YES is also specified on this parameter, the replacement character that is used each time an unprintable character is detected. Any printable EBCDIC character can be specified. Valid values range from 40 through 99 and A1 through FE.

ALIGN

Specifies whether the page must be aligned in the printer before printing is started. If ALIGN(*YES) and SPOOL(*NO) are specified, and forms alignment is required, the system sends a message to the message queue specified in the printer device description and waits for a reply to the message. When spool (*YES) is specified on the CRTPRTF command and ALIGN(*FILE) is specified on the STRPRTWTR command, then this parameter is used to determine whether an alignment message is sent by the system.

This parameter is ignored when cut sheets are used (spooled and direct output). Page alignment can be done only for text-only files. Page alignment cannot be done for print jobs containing graphics or bar codes.

This parameter overrides the alignment value specified in the printer file, in the program, or in other called OVRPRTF commands.

***NO:** No page alignment is required.

***YES:** The page is aligned before the output is printed.

DRAWER

Specifies the source drawer used when single-cut sheets are fed into the printer (specified by FORMFEED(*AUTOCUT)).

***E1:** The envelopes are fed from the envelope drawer on the sheet-feed paper handler.

***FORMDF:** The paper is fed from the source drawer specified in the form definition. If a form definition is not specified, then source drawer 1 is used.

source-drawer: Specify the drawer from which the paper is fed. Valid values range from 1 through 255.

OUTBIN

Specifies the destination of the output on printers capable of multiple output bins.

***DEVVD:** The destination of the output is the device default output bin.

output-bin: Specify the output bin for the destination of the output. Valid values range from 1 through 65535.

FONT Specifies the font identifier and point size used with this printer device file. If a font identifier or point size is not specified, the system automatically sets them.

More information about the valid font identifiers, the display value, the characters per inch value implied with each font style, a description of each font style, and whether the font is supported on

a particular printer is in the Printer Device Programming  book.

Note: Some fonts can be substituted by the printer. Consult the various printer reference guides for details.

***CPI:** The identifier of the font with the specified pitch (characters per inch (CPI)) is used.

***DEVVD:** The font identifier and point size specified in the device description are used.

Element 1: Font Identifier

identifier: Specify the numeric font identifier associated with this printer.

Element 2: Point Size

***NONE:** The point size is supplied by the system and is determined by the specified font identifier.

point-size: Specify a point size ranging from 0.1 through 999.9.

FORMFEED

Specifies the form feed attachment used by this printer device file.

***DEV D:** The forms are fed into the printer in the manner specified in the device description.

***CONT:** Continuous forms are used by the printer. The tractor feed attachment must be on the device.

***CONT2:** Continuous forms are used by the printer. The form is fed from the secondary tractor feed attachment. The secondary tractor feed attachment must be on the printer device.

***CUT:** Single-cut sheets are used by the printer. Each sheet must be manually loaded. For cut sheets, the forms alignment message is not sent.

***AUTOCUT:** The sheet-fed attachment must be on the printer. Single-cut sheets are automatically fed into the printer. The forms alignment message is not sent for cut sheets.

PRTQLTY

Specifies, for the 3812 SCS, 3816 SCS, 4214, 4224, 4230, 4234, and 5219 printers, the quality of print produced.

For the 5219 Printer, different print qualities are produced by varying the speed at which the print ribbon advances. Quality mode (*STD or *NLQ) results in normal print ribbon advancement. In draft mode (*DRAFT), the ribbon advances at a rate of one-third the distance it advances in quality mode. The 5219 Printer has a conserve ribbon switch that overrides the value of *DRAFT specified by this parameter.

For the 3812 SCS and 3816 SCS Printers, the automatic hardware selection of computer output reduction printing selected through soft switches on the printers occurs only when *DRAFT is specified for PRTQLTY and PAGRTT is *DEV D. If PAGRTT(*COR) is specified, the PRTQLTY parameter does not affect the printed output.

For the 4224, 4230, and 4234 Printers, standard print quality is produced by varying the density of the dot matrix pattern used to create printable characters. Standard mode (*STD) is the normal mode. Quality mode (*NLQ) requires multiple passes by the printer to produce a line of data. Draft mode (*DRAFT) results in high-speed printing.

For the 4214 printer, only draft (*DRAFT), quality (*NLQ), and device default (*DEV D) modes are supported. Other values are set to quality (*NLQ) mode.

More information about the valid values for the 4214, 4224, 4230, 4234, and 5219 Printers is in

the Printer Device Programming  book.

Notes:

1. For the 4214 Printer, quality mode (*STD or *NLQ) is only supported for 10 and 12 characters per inch. If PRTQLTY(*STD or *NLQ) and 5, 15, or 16.7 characters per inch is specified, the data is printed in draft mode.
2. For the 4234 Printer, only a limited character set (62 characters) is supported when PRTQLTY(*DRAFT) is specified. A description of the character set supported with draft print quality is in the *4234 Printer Operator's Guide*.
3. For the 4224 and 4230 printers, the fonts supported are not available for all three print qualities. The OCR-A and OCR-B fonts are supported only with PRTQLTY(*NLQ). The Courier and Essay fonts are available only with PRTQLTY(*NLQ) and PRTQLTY(*STD). The Gothic font is available only with PRTQLTY(*DRAFT) or PRTQLTY(*FASTDRAFT). If there is a mismatch between the print quality and the font selected, the font is changed to match the print quality.

4. Specify PAGRTT(*DEV) and PRTQLTY(*DRAFT) on this command to enable automatic rotation if the data does not fit on the paper.

***STD:** The output is printed with standard quality.

***DRAFT:** The output is printed with draft quality.

***DEV:** The print quality is set on the printer by the user, if it is not set within the data stream.

***NLQ:** The output is printed with near letter quality.

***FASTDRAFT:** The output is printed at a higher speed and with lower quality than it would be if you specified *DRAFT. This value is only supported by the 4230 printer.

CTLCHAR

Specifies whether the printer file supports input with print control characters. Any invalid control characters that are found are ignored, and single spacing is assumed.

***NONE:** No print control characters are passed in the data being printed.

***FCFC:** The first character of every record contains an American National Standards Institute (ANSI) forms control character. If *FCFC is specified, the record length must include one extra position for the first-character forms-control code. This value is not valid for externally described printer files.

***MACHINE:** The first character of every record contains a machine code control character. If *MACHINE is specified, the record length must include one extra position for the first character forms control code. This value is not valid for externally described printer files.

If TBLREFCHR(*YES) is also specified, then the record length must include two extra positions for the control character and the table reference character.

CHLVAL

Specifies a list of channel numbers with their assigned line numbers. Use this parameter only if CTLCHAR(*FCFC) has been specified.

Note:

If one or more channel-number/line-number combinations are changed, all other combinations must be re-entered.

***NORMAL:** The default values for skipping to channel identifiers are used. The default values are found in the following table.

Figure 1. ANSI First-Character Forms-Control Codes

Code	Action before Printing a Line
' '	Space one line (blank code)
0	Space two lines
-	Space three lines
+	Suppress space
1	Skip to line 1
2-11	Space one line
12	Skip to overflow line (OVRFLW parameter)

Element 1: Channel Number

channel-number: Specify an American National Standard channel number to be associated with a corresponding 'skip to' line number. Valid values for this parameter range from 1 through 12, corresponding to channels 1 through 12. The CHLVAL parameter associates the channel number with a page line number. For example, if you specify CHLVAL(2 20), channel identifier 2 is allocated with line number 20; therefore, if you place the forms-control 2 in the first position of a record, the printer skips to line 20 before printing the line.

Note:

If the printer stops and the next record processed has a channel value forms-control number that is the same value as the line number the printer is on, the printer advances to that value (line number) on the next page. However, if the printer is positioned at the top of the page (line number one) and the channel value forms-control value is associated with line number one, the printer does not advance to a new a new page.

If no line number is specified for a channel identifier, and that channel identifier is encountered in the data, a default of 'space one line' before printing is used. Each channel number can be specified only once.

Element 2: Line Number

line-number: Specify the line number assigned for the channel number in the same list. Valid line numbers range from 1 through 255. If no line number is assigned to a channel number, and that channel number is encountered in the data, a default of 'space one line' before printing is used. Each line number should be specified only once.

FIDELITY



Specifies whether printing continues when print errors are found for printers configured with AFP(*YES).

***CONTENT**: Printing continues when errors are found.

***ABSOLUTE**: Printing stops when errors are found.

CHRID

Specifies the character identifier (graphic character set and code page) for the file. This parameter allows printing of text that is in different character identifier (graphic character set and code page) coding. The value specified on this parameter is used to instruct the printer device to interpret the hexadecimal byte string to print the same characters that were intended when the text was

created. More information about the character identifier is in the Printer Device Programming  book. A list of valid CHRID values and applicable printers is in the "CHRID Values and Applicable Printers (CHRID parameter)" table in Printer Device Programming  book.

***DEV**: The default CHRID value that the device is designed to handle is used. The *DEV value means character selection is normal because the file has the same character identifier as the device default.

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

***JOBCCSID**: The character identifier for the printer file is taken from the coded character set identifier (CCSID) of the job.

Note:

This value is not allowed if the file was created on a system at an earlier release level than V2R3M0.

***CHRIDCTL:** The system checks the CHRIDCTL job definition attribute to determine whether to use *JOBCCSID or *DEVD on the CHRID command parameter for this file.

Element 1: Character Set

graphic-character-set: Specify the graphic character set values that match the attributes of the printer. Valid values range from 1 through 32767.

Element 2: Code Page

code-page: Specify the code page value that matches the attributes of the printer. Valid values range from 1 through 32767.

DECFMT

Specifies which decimal format value is used when editing numeric fields with the EDTCDE DDS keyword. The decimal format value determines the use of commas and periods for the decimal position and three digit positional separators on edited fields.

***FILE:** Use the decimal format value stored with the file when the file was created.

***JOB:** Use the decimal format value from the DECFMT job attribute when the file is opened.

FNTCHRSET

Specifies a downloaded font consisting of a character set and code page. For an outline font, a point size is required. For a raster font, the point size is ignored. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

***FONT:** The value specified on the FONT parameter is used.

Element 1: Font Character Set

The name of the font character 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.

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

character-set: Specify the font character set to use.

Element 2: Code Page Name

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

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

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

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

code-page: Specify the code page value used to create the command parameters. Valid values range from 1 through 999.

Element 3: Point Size

***NONE**: The point size is supplied by the system and is determined by the specified font identifier.

point-size: Specify a point size ranging from 0.1 through 999.9.

CDEFNT

Specifies the coded font that the system uses for single-byte character set (SBCS) printing. For coded fonts that reference an outline font, a point size may also be specified. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

***FNTCHRSET**: The font specified on the FNTCHRSET parameter is used.

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

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

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

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

coded-font-name: Specify the DBCS-coded font name to use.

Element 2: Point Size

***NONE**: The point size is supplied by the system and is determined by the specified font identifier.

point-size: Specify a point size ranging from 0.1 through 999.9.

PAGDFN

Specifies the qualified name of the page definition to be used to format line data.

***NONE**: No page definition is specified.

Because PSF/400 requires a page definition when *LINE or *AFPSDLINE is specified, an inline page definition is built from the print file parameters and passed to PSF/400 when *NONE is specified.

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

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

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

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

page-definition-name: Specify the name of the page definition that must exist in the library specified. Valid values range from 1 to 8 characters. Device type *AFPDSLIN or *LINE must be specified when using a page definition.

FORMDF

Specifies the form definition to use when printing the file. A form definition is a resource object that defines the characteristics of the form, including overlays, position of page data on the form, and number of copies of pages and modifications to pages. The form definition is located inline with the file being printed, or in a library.

***NONE**: No form definition is used.

Because PSF/400 requires a form definition, an inline form definition is built from the print file parameters and passed to PSF/400 when *NONE is specified.

***DEVD**: The name of the form definition is specified in the printer device description.

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.

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

form-definition-name: Specify the name of the form definition that must exist in the library specified. Valid values range from 1 to 8 characters.

AFPCHARS

Specifies one or more AFP characters (coded fonts) to be used with line data and a page definition.

***NONE**: No AFP character (coded fonts) specified.

coded-font-name: Specify up to four 4-byte names of coded fonts to be specified with line data and a page definition. The 4-byte names would be concatenated to X0 to identify up to four coded fonts which are to be used when TBLREFCHR is being used within the data.

TBLREFCHR

Specifies whether table reference characters are present in the line data.

***NO**: No table reference character is present in line data.

***YES**: Table reference characters are present in line data.

If forms control characters are used with the data, the table reference character follows the forms control character but precedes the data bytes. If forms control characters are not used, the table reference character is the first byte of the data record. As with forms control character, if table reference characters are used, every data record must contain a TRC byte.

PAGRRT

Specifies the degree of text rotation for the 3112, 3116, 3130, 3812, 3816, 4028, 3820, 3825, 3827, 3829, 3831, 3835, 3900, 3916, 3930 and 3935 printers. This parameter allows the user to specify the degree of rotation of the text on the page with respect to the way the form is loaded into the printer. See the note under the PAGESIZE parameter for directions on specifying page size when rotating the page.

Specify *AUTO or *DEVD for this parameter and PRTQLTY(*DRAFT) on this command to enable automatic rotation if the data does not fit on the paper.

***AUTO:** Indicates that automatic rotation of output is done to fit the printed data on the form. If rotation does not accomplish this, computer output reduction is performed automatically (regardless of the print quality being used). This parameter is valid only for printers supporting rotation.

***DEVD:** The operating system sends a device default rotation value to the printer. Page rotation is dependent on your printer's specifications. See your printer or printer emulation documentation to determine how page rotation is affected.

***COR:** Computer output reduction is done. Computer output reduction allows printed output intended for a 13.2 inch wide by 11.0 inch long form to be printed on an 11 inch wide by 8.5 inch long form.

For computer output reduction printing, the following operations are done for the 3112, 3116, 3130, 3812, 3816, 4028, 3820, 3825, 3827, 3829, 3831, 3835, 3900, 3916, 3930 and 3935 printers:

- Automatic rotation to *COR is not done if the file contains graphics, bar codes, variable LPI, variable font, variable page rotations, or variable drawer.
- The text is rotated 90 degrees clockwise from the 0 degree rotation position (lower left corner of the first edge loaded into the printer).

Note:

For landscape paper on a 3835 printer, the rotation is counter-clockwise from the 0 degree rotation position (upper right corner of the first edge loaded into the printer).

- A top and left margin of 0.5 inches is added to the printed output.
- The 12-pitch fonts are changed to a 15-pitch font and 15-pitch fonts are changed to a 20-pitch font. All other font widths are changed to a 13.3-pitch font, except for the 4028 printer where they are changed to a 15-pitch font.
- Vertical spacing (specified by the LPI parameter) is 70 percent of the normal spacing.
- The page size is set to 8.5 inches wide by 11 inches long.

0: The format of this tape is QIC3040, which is used for 1/4 inch minicartridge tapes that can hold up to 840 megabytes of data.

90: Rotation of the text is done 90 degrees clockwise from the 0 degree writing position.

180: Rotation of the text is done 180 degrees clockwise from the 0 degree writing position.

270: Rotation of the text is done 270 degrees clockwise from the 0 degree writing position.

MULTIUP

Specifies, for spooled output only, the number of pages printed on a single physical page.

Note:

Overlays are not reduced when more than one page is printed on a side.

For examples and more details see the Printer Device Programming  book.

1: One page of output is printed on one physical sheet of paper.

2: Two pages of output are printed on 1 physical sheet of paper.

3: Three pages of output are printed on 1 physical sheet of paper.

4: Four pages of output are printed on 1 physical sheet of paper.

REDUCE

Specifies whether or not to reduce the output when doing multiple up printing.

For examples and more details see the Printer Device Programming  book.

***TEXT:** The text output is reduced when doing multiple up printing.

***NONE:** The output is not reduced when doing multiple up printing.

PRTTXT

Specifies up to 30 characters of text to be printed at the bottom of each page of output. More information on this parameter is in Commonly used parameters.

***JOB:** The value for the current job is used.

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

'print-text': Specify the character string printed at the bottom of each page. No more than 30 characters of text can be entered, enclosed in apostrophes.

JUSTIFY

Specifies the printing positions of the characters on a page so the right-hand margin of printing is regular. Justification is done to the record length on the printer file opened.

Note:

The JUSTIFY parameter is supported only on the 3812 SCS, 3816 SCS, and 5219 Printers.

0: 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.

50: Spaces are added to the blanks in the text so that the right margin is more closely aligned but not flush.

100: The text is expanded by spaces (added where the blanks already exist) until the right margin is flush.

DUPLEX

Specifies whether output is printed on one side or two sides of the paper.

***NO:** The output is printed on one side of the paper.

***YES:** The output is printed on both sides of the paper with the top of each printed page at the same end of the paper.

***TUMBLE:** The output is printed on both sides of the paper with the top of one printed page at the opposite end of the sheet from the top of the other printed page. This is usually used for output that is bound at the top.

***FORMDF:** The output is printed on both sides of the paper if the duplex value is specified in the form definition. If a form definition is not specified, then the output is printed on one side of the paper.

UOM Specifies the unit of measure that is used.

***INCH:** An inch is used as the unit of measure.

***CM:** A centimeter is used as the unit of measure.

FRONTOVL

Specifies the qualified name of the object that contains both the overlay that is printed on the FRONT side of the page and the offset, down and across, from the point of origin used when the overlay is printed.

***NONE:** No overlay is used.

Element 1: Overlay Name

The name of the overlay 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.

overlay-name: Specify the name of the overlay.

Element 2: Offset Down

0: No offset down from the point of origin is used.

offset-down: Specify the offset down from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

Element 3: Offset Across

0: No offset across from the point of origin is used.

offset-across: Specify the offset across from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

BACKOVL

Specifies the object name and library name containing both the overlay that is printed on the BACK side of the page and the offset, down and across, from the point of origin used when the overlay is printed.

***FRONTOVL:** The values that are specified on the FRONTOVL parameter are used.

***NONE:** No overlay is used.

Element 1: Overlay Name

The name of the overlay 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.

overlay-name: Specify the name of the overlay.

Element 2: Offset Down

0: No offset down from the point of origin is used.

offset-down: Specify the offset down from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

Element 3: Offset Across

0: No offset across from the point of origin is used.

offset-across: Specify the offset across from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

Element 4: Constant Back

The constant back function allows you to print overlays on blank pages without adding blank pages to the print application. Specifying the constant back function would cause, for each page generated by the application program, a blank page to be generated onto which the specified back overlay could be printed. The generated blank pages are called constant forms because no variable data from the user's program is printed on the pages. The constant back function is only supported for duplex printing. It is ignored when DUPLEX(*NO) is specified on the printer file.

Note that the offset down and offset across values are ignored when *CONSTANT is specified for constant back. An offset of 0.0 is assumed for these values.

***NOCONSTANT**: No constant back is specified.

***CONSTANT**: Constant back is specified.

CVTLINDTA

Specifies whether line data is converted to Advanced Function Presentation Data Stream (AFPDS) before the data is written to the spooled file. When DEVTYPE(*LINE) or DEVTYPE(*AFPDSL) is specified, and a page definition is specified (PAGDFN Parameter), this parameter allows the line data to be converted to AFPDS before the data is written to spooled file. For device types of *SCS,*USERASCII, *IPDS, and *AFPDS, this parameter is ignored. For device types of *LINE and *AFPDSL, if a page definition is not specified, then this parameter is ignored.

To print AFPDS spooled files on an OS/400 requires Host Print Transform when printing to ASCII attached printers and PSF/400 (optional feature of OS/400) for IPDS attached printers.

***NO**: Line data is not converted to AFPDS.

***YES**: Specifies that line data is converted to AFPDS before the data is written to the spooled file.

IPDSPASTHR

Specifies whether IPDS (intelligent printer data stream) pass-through is done for the spooled file.

***DEV**: The value specified for IPDSPASTHR in the PSF configuration object specified for a printer device description is used. If no PSF configuration object is specified for the device, a value of *NO is used.

***NO:** No IPDS pass-through is done.

***YES:** Specifies that IPDS pass-through is to be done if the spooled file is eligible for IPDS pass-through.

Note:

Not all SCS or IPDS spooled files are eligible for IPDS pass-through. They may contain special functions that require transform to AFPDS for correct printing. Specifying IPDS pass-through on the printer file allows only those spooled files eligible for IPDS pass-through to bypass the extra transforms. Those spooled files not eligible for IPDS pass-through will still undergo the transforms to AFPDS and back to IPDS.

IPDS pass-through will not be valid for all PSF/400 supported printers. Any printer (or attachment) that does not support resident fonts can not support IPDS pass-through. This is because the resident font references in the data stream must be mapped to host fonts which are downloaded to the printer. All IBM IPDS printers, except for the following, can be supported with IPDS pass-through: 3820, 3825, 3827, 3828, 3829, 3831, 3835, 3900-001 and any printer attached using Print Services Facility for OS/2's Distributed Print Function.

For V3R7, V4R1 and V4R2, IPDSPASTHR can be specified with the USRDFNDDTA parameter in a printer file. You may continue using this support with existing printer files and PSF configuration objects by specifying IPDSPASTHR(*DEVDD) in the printer file. If you specify a value of anything other than *DEVDD for the IPDSPASTHR parameter, any IPDS pass-through value in the USRDFNDDTA parameter is ignored.

USRRSCLIBL

Specifies the list of user resource libraries to be used for searching for AFP resources for a spooled file. If the AFP resource is not found in the user resource libraries, then the library list specified in the DEVRSCLIBL parameter of the PSF configuration object is searched. If no PSF configuration object is specified for the device, then libraries QFNTCPL, QFNT01-QFNT19, and QFNT61-69 are searched.

***DEVDD:** The value specified for USRRSCLIBL in the PSF configuration object specified for a printer device description is used. If no PSF configuration object is specified for the device, a value of *JOBLIBL is used.

***NONE:** No user libraries are specified.

***JOBLIBL:** Specifies that the library list of the job that created the spool file is used in searching for AFP resources. This library list is saved with the spool file when it is created.

***CURLIB:** Specifies that the current library of the job that created the spool file is used for searching for AFP resources. If no library is specified as the current library for the job, then library QGPL is used.

user-resource-library-name: Specify the name of a library that will be used to search for AFP resources. Up to four library names may be specified.

For V3R7, V4R1 and V4R2, USRRSCLIBL can be specified with the USRDFNDDTA parameter in a printer file. PSF/400 uses that value if USRRSCLIBL(*PRTF) is specified in a PSF configuration object which is specified in the printer device description. You may continue using this support with existing printer files and PSF configuration objects by specifying USRRSCLIBL(*DEVDD) in the printer file. If you specify a value of anything other than *DEVDD for the USRRSCLIBL parameter, any user resource library value in the USRDFNDDTA parameter is ignored.

CORNERSTPL

Specifies the reference corner to be used for a corner staple. A staple is driven into the media at the reference corner. Refer to your printer's documentation for information as to which reference corners are supported.

Page rotation does not affect the placement of a corner staple.

***NONE:** A corner staple is not specified.

***DEVD:** The reference corner is the default reference corner used by the device.

***BOTRIGHT:** The reference corner is the bottom right corner of the media.

***TOPRIGHT:** The reference corner is the top right corner of the media.

***TOPLEFT:** The reference corner is the top left corner of the media.

***BOTLEFT:** The reference corner is the bottom left corner of the media.

EDGESTITCH

Specifies the placement of staples along the finishing margin in either inches or centimeters (specified in the unit of measure (UOM) field). The finishing margin can be thought of as an imaginary line parallel to the edge of the paper where the staples will be placed. The position of the finishing margin relative to the physical edge is set in

Page rotation does not affect the placement of an edge stitch.

Single Value

***NONE:** An edge stitch is not specified.

Element 1: Reference Edge

Specifies the reference edge to be used for an edge stitch. An edge stitch is formed by having one or more staples driven into the media along the finishing operation axis.

***DEVD:** The reference edge is the default reference edge used by the device.

***BOTTOM:** The reference edge is the bottom edge of the media.

***RIGHT:** The reference edge is the right edge of the media.

***TOP:** The reference edge is the top edge of the media.

***LEFT:** The reference edge is the left edge of the media.

Element 2: Reference Edge Offset

Specifies the offset of the edge stitch from the reference edge toward the center of the media.

***DEVD:** The reference edge offset is the default reference edge offset used by the device.

reference-edge-offset: Specify the offset of the edge stitch from the reference edge. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This value is converted to millimeters for the printer. Fractional millimeters are not supported and are discarded when conversion to millimeters is performed.

Element 3: Number of Staples

Specifies the number of staples that are to be applied along the finishing operation axis.

***DEVD:** The number of staples depends on the value of the Staple Offsets element of this parameter. If *DEVD is also specified or defaulted for the Staple Offsets element value, then the number of staples is the default number of staples used by the device. If one or more offsets are specified for Staple Offsets, the number of staples is the same as the number of staple offsets specified.

number-of-staples: Specify the number of staples to be used for the edge stitch. Valid values range from 1 to 122 staples. If one or more offsets are specified for Staple Offsets, the number of staples is the same as the number of staple offsets specified.

Element 4: Staple Offsets

Specifies the offset of the staples along the finishing operation axis. The offset is measured from the point where the finishing operation axis intersects either the bottom edge or the left edge of the media, toward the center of the media. Each consecutive value is used to position a single finishing operation centered on the specified point on the finishing operation axis.

***DEV**: The staple offsets are the default staple positions used by the device. If a value was specified for the Number of Staples element, the staple position of each staple will be calculated automatically by the printer.

staple-offset: Specify the staple offset for each staple in the edge stitch. Up to 122 staple offsets may be specified. If one or more offsets are specified, and a value was specified for Number of Staples, the number of staple offsets will take precedence. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This value is converted to millimeters for the printer. Fractional millimeters are not supported and are discarded when conversion to millimeters is performed.

SADLSTITCH

Specifies where one or more staples are driven into the media along the finishing operation axis, which is positioned at the center of the media parallel to the reference edge.

Page rotation does not affect the placement of a saddle stitch.

Single Value

***NONE**: A saddle stitch is not specified.

Element 1: Reference Edge

Specifies the reference edge to be used for a saddle stitch. A saddle stitch is formed by having one or more staples driven into the media along the finishing operation axis, which is positioned at the center of the media parallel to the reference edge.

***DEV**: The reference edge is the default reference edge used by the device.

***TOP**: The reference edge is the top edge of the media.

***LEFT**: The reference edge is the left edge of the media.

Element 2: Number of Staples

Specifies the number of staples that are to be applied along the finishing operation axis.

***DEV**: The number of staples depends on the value of the Staple Offsets element of this parameter. If ***DEV** is also specified or defaulted for the Staple Offsets element value, then the number of staples is the default number of staples used by the device. If one or more offsets are specified for Staple Offsets, the number of staples is the same as the number of staple offsets specified.

number-of-staples: Specify the number of staples to be used for the saddle stitch. Valid values range from 1 to 122 staples. If one or more offsets are specified for Staple Offsets, the number of staples is the same as the number of staple offsets specified.

Element 3: Staple Offsets


Specifies the offset of the staples along the finishing operation axis. The offset is measured from the point where the finishing operation axis intersects either the bottom edge or the left edge of the media, toward the center of the media. Each consecutive value is used to position a single finishing operation centered on the specified point on the finishing operation axis.

***DEV D:** The staple offsets are the default staple positions used by the device. If a value was specified for the Number of Staples element, the staple position of each staple will be calculated automatically by the printer.

staple-offset: Specify the staple offset for each staple in the saddle stitch. Up to 122 staple offsets may be specified. If one or more offsets are specified, and a value was specified for Number of Staples, the number of staple offsets will take precedence. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This value is converted to millimeters for the printer. Fractional millimeters are not supported and are discarded when conversion to millimeters is performed.

FNTRSL

Specifies the resolution PSF/400 uses when printing to a multiple resolution printer configured to report multiple resolutions, but the spooled file does not specify the font metrics and resolution or the font is not available at the resolution that is contained in the spooled file.

For more information regarding the algorithm used for searching a library list for a font resource, see the Printer Device Programming  book section entitled User and Device Resource Library Lists in the chapter called Working With PSF configuration objects.

***DEV D:** The value specified in the FNTRSL parameter of the PSF configuration object for the device is used. If no PSF configuration object is specified for the device, a value of *SEARCH is used.

***SEARCH:** Specifies to search the library list for the first occurrence of a host font with a name match. The resolution of that font is used to print the spool file. Message PQT3546 is sent to specify the resolution of the font that was selected.

240: The font resolution is 240 pels per inch.

300: The font resolution is 300 pels per inch.

DFRWRT

Specifies how much output is held in the system buffer before being sent to the printer.

Note:

If this command is specified, the Display Override (DSPOVR) command will either display or print the override along with any others that are specified on this command.

***YES:** The system controls the amount of output that is held in the buffer before being sent to the printer.

If SPOOL(*YES) is specified along with SCHEDULE(*IMMED), output is held in the buffer until a page of output is available or until the system buffer is full.

***NO:** If SPOOL(*NO) is specified, output is not held in the buffer. Output is sent to the printer immediately after the program performs a write operation.

If the spooled output schedule is not immediate, specifying DFRWRT(*NO) has no effect.

SPOOL

Specifies whether the output data for the printer file is spooled. If SPOOL(*NO) is specified, the following parameters in this command which only apply to spooled files are ignored: OUTQ, COPIES, PAGERANGE, MAXRCDS, FILESEP, SCHEDULE, HOLD, SAVE, OUTPTY, USRDTA, SPLFNAME, SPLFOWN, USRDFNOPT, USRDFNDDTA, and USRDFNOBJ. In addition, several other parameters in this command are not supported for SPOOL(*NO) because they either require PSF/400 or are only supported for certain device types which cannot be specified with

SPOOL(*NO). These parameters are: FRONTMGN, BACKMGN, FIDELITY, FNTCHRSET, CDEFNT, PAGDFN, FORMDF, AFPCHARS, TBLREFCHR, REDUCE, FRONTOVL, BACKOVL, IPDSPASTHR, USRRSCLIBL, CORNERSTPL, EDGESTITCH, SADLSTITCH, FNTRSL, and CVTLINDTA.

Note:

If SPOOL(*NO) is the current value in the printer file, or if SPOOL(*NO) is specified in this or any other OVRPRTF command that is in effect when the file is opened, the parameters that apply to spooled files are ignored. This parameter overrides the spool value specified in the printer file, or in other called OVRPRTF commands.

***YES:** The data is spooled for processing by a diskette writer or a print writer.

***NO:** The data is not spooled; it is sent directly to the device and printed as the output becomes available.

OUTQ Specifies, for spooled output only, the qualified name of the output queue. This parameter overrides the output queue name specified in the printer file or in other called OVRPRTF commands.

***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.

***JOB:** The output queue associated with the job is used.

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 to which the output data is spooled.

FORMTYPE

Specifies the type of form on which the output is printed. The identifiers used to indicate the type of forms are user-defined and can be a maximum of 10 characters in length.

Note:

If a form type other than *STD is specified, the system (when the output is produced) sends a message that identifies the form type to the system operator, and requests that the specified type of forms be put in the printer. This parameter overrides the form type value specified in the printer file or in other called OVRPRTF commands.

***STD:** The standard form type is used. The output is printed on the form type specified in the printer file for the printers selected. The printer file contains information that controls how the document is printed on a particular printer.

form-type: Specify the identifier of the form type used with this device file for printed output from jobs. Up to 10 alphanumeric characters can be specified. When the device file is opened, the system sends a message identifying the form type to the system operator, and requests that the identified forms be in the printer.

COPIES

Specifies, for spooled files, the number of copies being printed.

Note:

This parameter overrides the copy value specified in the printer file.

number-of-copies: Specify a value, ranging from 1 through 255, that indicates the number of identical printouts produced when this printer file is used.

PAGERANGE

Specifies the page range to print for each copy of the file to be printed.

Element 1: Starting Page to Print

***ENDPAGE**: Only the ending page is printed.

starting-page: Specify the page on which to start printing.

Element 2: Ending Page to Print

***END**: The last page in the file is printed.

ending-page: Only the ending page is printed.

MAXRCDS

Specifies, for spooled output only, the maximum number of records that can be in the spooled file for jobs using this printer file. If this maximum is reached, an inquiry message is sent to the program message queue. This parameter overrides the value specified in the printer file or in other called OVRPRTF commands.

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

maximum-records: Specify a value, ranging from 1 through 999999, that specifies the maximum number of records allowed in the spooled file.

FILESEP

Specifies, for spooled output only, the number of separator pages placed at the start of each printed file, including those between multiple copies of the same output. Each separator page has the following items printed on it: file name, file number, job name, user name, and the job number. This parameter overrides the separator value specified in the printer file or in other called OVRPRTF commands.

number-of-file-separators: Specify the number of separator pages used at the start of each printed output file produced by this device file. Valid values range from 0 through 9. If 0 is specified, no separator pages are printed for the file. In this case, the printed output for each file (or copy of a file) starts at the top of a new page.

SCHEDULE

Specifies, for spooled output only, when the spooled file is available to a writer. This parameter overrides the scheduling value specified in the printer file or in other called OVRPRTF commands.

***JOBEND**: The spooled file is made available to the writer only after the entire job is completed.

***FILEEND**: The spooled file is made available to the writer as soon as the file is closed in the program.

***IMMED:** The spooled file is made available to the writer as soon as the file is opened in the program.

HOLD Specifies, for spooled output only, whether the spooled file is held. The spooled file can be released by using the Release Spooled File (RLSSPLF) command.

Note: This parameter overrides the hold value specified in the printer file or in other called OVRPRTF commands.

***NO:** The spooled printer file is not held by the output queue. The spooled output is available to a writer based on the SCHEDULE parameter value.

***YES:** The spooled file is held until released by the Release Spool File (RLSSPLF) command.

SAVE Specifies, for spooled output only, whether the spooled file is saved (left on the output queue) after the output has been produced. This parameter overrides the save value specified in the printer file or in other called OVRPRTF commands.

***NO:** The spooled file data is not saved on the output queue after it has been produced.

***YES:** The spooled file data is saved on the output queue until the file is deleted. After the file is produced, the number of copies (see COPIES parameter) is set to 1, and its status is changed from WTR to SAV. Refer to the Release Spooled File (RLSSPLF) command for information on how to produce the spooled file again.

OUTPTY

Specifies the output priority for spooled output files that are produced by this job. The highest priority is 1 and the lowest priority is 9. More information on this parameter is in Commonly used parameters.

***JOB:** The output priority associated with the job that created the spooled file is used.

output-priority: Specify the output priority. Valid values range from 1 (high priority) through 9 (low priority).

USRDTA

Specifies, for spooled output only, the user-specified data that identifies the file.


***SOURCE:** If the spooled file was created by an application program, the name of the program is used. Otherwise, blanks are used.

user-data: Specify up to 10 characters of text.

SPLFOWN

Specifies, for spooled output only, who the owner of the spooled file is.


***CURUSRPRF:** The spooled file is owned by the current effective user of the current job or thread.

See the Printer Device Programming  book for information on how the SPLFOWN parameter is affected when using any of the following APIs:

- QWTSETP - Set Profile
- qsyssetuid() - Set User ID
- qsysseteuid() - Set Effective User ID
- qsyssetreuid() - Set Real and Effective User ID

***JOB:** The spooled file is owned by the original user profile of the job. If the job has switched to a new user profile, the original user profile is still the owner of the spooled file.

***CURGRPPRF:** The spooled file is owned by the current effective group profile of the current job or thread. If there is no current effective group profile, ownership of the spooled file is determined

in the same manner as *CURUSRPRF. See the Printer Device Programming  book for information on how the SPLFOWN parameter is affected when using any of the following APIs:

- QWTSETP - Set Profile
- qsysetgid() - Set Group ID
- qsysetegid() - Set Effective Group ID
- qsysetregid() - Set Real and Effective Group ID

***JOBGRPPRF:** The spooled file is owned by the group profile of the original user profile of the job. If the job has switched to a new user profile, the group profile of the original user profile is still the owner of the spooled file. If no group profile exists, ownership of the spooled file is determined the same way as *JOB.

USRDFNOPT

Specifies, for spooled output only, one or more user-defined options to be used by user applications or user-specified programs that process spooled files. A maximum of four user-defined options can be specified. This parameter overrides the user-defined options specified in the printer file or in other called OVRPRTF commands.

***NONE:** No user-defined options specified.

user-defined-option: Specify the user-defined option to be used by user applications or user-specified programs that process spooled files. All characters are acceptable.

USRDFNDDTA

Specifies, for spooled output only, the user-defined data to be used by user applications or user-specified programs that process spooled files. This parameter overrides the user-defined data specified in the printer file or in other called OVRPRTF commands.

***NONE:** No user-defined data specified.

user-defined-data: Specify the user-defined data to be used by user applications or user-specified programs that process spooled files. All characters are acceptable.

USRDFNOBJ

Specifies, for spooled output only, the qualified name and type of the user-defined object to be used by user applications or user-specified programs that process spooled files. This parameter overrides the user-defined object name specified in the printer file or in other called OVRPRTF commands.

***NONE:** No user-defined object specified.

Element 1: Name of User-Defined Object

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

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

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

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

object-name: Specify the user-defined object to be used by user applications or user-specified programs that process spooled files.

Element 2: Type of User-Defined Object

object-type: The user object type can be one of the following:

***DTAARA**

Data Area

***DTAQ**

Data Queue

***FILE** File

***PSFCFG**

PSF Configuration Object

***USRIDX**

User Index

***USRQ**

User Queue

***USRSPC**

User Space

SPLFNAME

Specifies, for spooled output only, the spooled file name.

***FILE**: The name of the printer file is used for the spooled file name.

spooled-file-name: Specify up to 10 characters for the spooled file name.

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.

This parameter overrides the wait time specified in the printer file, in the program, or in other called OVRPRTF commands.

***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 printer file 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.

LVLCHK

Specifies whether the record format level identifiers in the program are checked against those in the device file when the file is opened. If so, the record format identifiers in the program must

match those in the device file. Because the same record format name can exist in more than one file, each record format is given an internal system identifier when it is created.

Note: This parameter overrides the value specified in the printer file, in the program, or in other called OVRPRTF commands. Level checking cannot be done unless the program contains the record format identifiers. This command cannot override level checking from *NO to *YES.

***NO:** The level identifiers are not checked when the file is opened.

SECURE

Specifies whether this file is safe from the effects of previously called file override commands. If SECURE is not specified, processing occurs as if SECURE(*NO) is specified.

***NO:** This file is not protected from the effects of other file overrides; its values can be overridden by the effects of previously called file override commands.

***YES:** This file is protected from the effects of any file override commands previously called.

OVRSCOPE

Specifies the extent of influence (scope) of the override.

***ACTGRPDFN:** The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL:** The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB:** The scope of the override is the job in which the override occurs.

SHARE

Specifies whether the open data path (ODP) for the printer 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.

***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.

Note: 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, 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.

OPNSCOPE

Specifies the extent of influence (scope) of the open operation.

***ACTGRPDFN:** The scope of the open data path (ODP) is determined by the activation group of the program that called the OVRPRTF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller. In a multi-threaded job, only those shared opens within the same thread and the same activation group can share this ODP.

***JOB:** The scope of the open data path (ODP) is the job in which the open operation occurs. If the job is multi-threaded, only those opens from the same thread can share this ODP.

IGCDTA

Specifies, for program-described original files, whether the file processes double-byte character set (DBCS) data. For externally described printer files, this parameter specifies DBCS attributes of the file.

For program-described files:

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

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

IGCEXNCHR

Specifies whether the system processes double-byte character set (DBCS) extension characters.

***YES:** The system processes DBCS extension characters.

***NO:** The system does not process DBCS extension characters; it prints extension characters as the undefined character.

IGCCHRTT

Specifies, for the 5553 and 5583 Printers only, whether the printer rotates double-byte characters 90 degrees counterclockwise when printing. The system prints rotated double-byte characters so they appear in a vertical reading sequence. Alphanumeric characters are not rotated.

***NO:** The system does not rotate double-byte characters when printing.

***YES:** The system rotates double-byte characters 90 degrees counterclockwise when printing. The printer rotates each character individually.

IGCCPI

Specifies the printer character density of double-byte character set (DBCS) characters, in characters per inch (CPI).

Note:

This parameter does not specify the printer character density of alphanumeric characters. Alphanumeric characters are printed with the value specified on the CPI parameter.

***CPI:** DBCS character density is based on the values specified for the CPI parameter. The system prints one double-byte character for every two alphanumeric characters.

- For CPI(10), DBCS characters print at 5 characters per inch.
- For CPI(12), DBCS characters print at 6 characters per inch.
- For CPI(13.3), DBCS characters print at 6.7 characters per inch (same as IGCCPI(*CONDENSED)).
- For CPI(15), DBCS characters print at 7.5 characters per inch.
- For CPI(18), DBCS characters print at 9 characters per inch.
- For CPI(20), DBCS characters print at 10 characters per inch.

***CONDENSED:** Condensed printing is used in which the system prints 20 DBCS characters every 3 inches. This value is valid only for the 5553 or 5583 Printers.

5: The format of this tape is QIC525, which is used for 1/4 inch cartridge tapes that can hold 525 megabytes of data.

6: DBCS character density is 6 characters per inch. This value is valid for the 5553 and 5583 Printers only.

10: DBCS character density is 10 characters per inch. This value is valid for the 5553 or 5583 Printers only.

IGCSOSI

Specifies, for bracketed DBCS character strings only, how the system prints shift control characters.

***NO:** The system does not print shift control characters. These characters do not occupy a position in printed output.

***YES:** The system prints shift control characters as blanks.

***RIGHT:** The system prints two blanks when printing shift-in characters but does not print shift-out characters.

IGCCDEFNT

Specifies the coded font that the system uses for DBCS printing. For a coded font that references an outline font, a point size may also be specified.

***SYSVAL:** The DBCS-coded font specified in the system value QIGCCDEFNT is used.

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

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

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

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

coded-font-name: Specify the coded font name to use.

Element 2: Point Size

***NONE:** The point size is supplied by the system and is determined by the specified font identifier.

point-size: Specify a point size ranging from 0.1 through 999.9.

Examples for OVRPRTF

Example 1: Printing Output

```
OVRPRTF FILE(PRINTOUT) TOFILE(PRINT3) SPOOL(*YES)
      COPIES(5) OUTQ(OUTPUT1)
```

This command overrides the file named PRINTOUT and uses the printer file named PRINT3 to produce the spooled output on the printer. The output from the program is sent to the OUTPUT1 output queue. Five copies of the spooled file are printed on the printer specified on the Start Printer Writer (STRPRTWTR) command.

Example 2: Rotating Double-Byte Characters

```
OVRPRTF FILE(IGCLIB/IGCPRT) IGCDTA(*YES)
      IGCHRRRT(*YES)
```

This command overrides the IGCPRT printer file, which is stored in the IGCLIB library. The override puts the IGALTTYP DDS keyword into effect to change character output fields to DBCS fields, and rotates the double-byte characters when printing.

Error messages for OVRPRTF

*ESCAPE Messages

CPF180C

Function &1 not allowed.

CPF7343

Channel number specified more than once on CHLVAL.

OVRSAVF (Override with Save File) Command Description

OVRSAVF Command syntax diagram

Purpose

The Override with Save File (OVRSAVF) command is used to (1) override or replace a file named in a program, (2) override certain attributes of a file that are used by a program, or (3) override the file named in a program and certain attributes of the overriding file.

Note: Overrides do not apply to save and restore commands.

More information on overriding files is in the File Management topic in the Information Center.

Required Parameter

FILE Specifies the name of the file being used by the program to which this override command is applied. If TOFILE(*FILE) is specified, a display device file must be specified. Otherwise, any device file or database file can be specified.

Note: The information in a save file has meaning only to Operating System/400 save and restore; redirecting another type of file to a save file or vice versa is not recommended.

Optional Parameters

TOFILE

Specifies the name of the save file that is used instead of the file specified in the FILE parameter or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified on this command. The parameters specified on this command override the other values specified in the save file or in the program.

***FILE:** The save file named in the FILE parameter has certain parameters overridden by the values specified in this command.

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

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

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

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

save-file-name: Specify the qualified name of the save file that is used instead of the overridden file name. If no library qualifier is given, *LIBL is used to find the save file.

EXTEND

Specifies, for output operations only, whether new records are added to the end of the data currently in the save file. This option is used to start processing after an application or system failure. When this operation is completed, the file must contain the image of a single save operation made by a save command, or it may not be possible to restore objects from the save file. This parameter overrides the extend value specified in the program. The sequencing information in the file's records guarantees that after a system failure, a record cannot be skipped or sent twice.

***NO:** Records are not added to the end of the specified save file, but they replace existing records in the file. If the save file already contains records, an inquiry message is sent that clears the file or cancels the operation. If no EXTEND value is specified by the program or in an override, this is the default action assumed when the file is opened for output.

***YES:** New records are added to the end of the records already contained in the save file.

POSITION

Specifies the starting position for getting records from the save file. The first record to get is either at the beginning of the file (*START) or at a particular relative record number position in the file (*RRN). This parameter overrides the value specified in the program.

***START:** Get the first record in the file first. If no POSITION value is specified by the program, or in an override, this is the default action assumed when the file is opened for input.

***RRN** *relative-record-number:* Specify the record number (its position from the beginning of the file) of the record that the user gets first. The value *RRN must go before the relative record number. For example, POSITION(*RRN 480) specifies that the 480th record in the file is gotten first.

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.

This parameter overrides the wait time specified in the program or in the save 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.

SECURE

Specifies whether this file is safe from the effects of previously called file override commands. If SECURE is not specified, processing occurs as if SECURE(*NO) is specified.

***NO:** This file is not protected from the effects of other file overrides; its values can be overridden by the effects of previously called file override commands.

***YES:** This file is protected from the effects of any file override commands previously called.

OVRSCOPE

Specifies the extent of influence (scope) of the override.

***ACTGRPDFN:** The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL:** The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB:** The scope of the override is the job in which the override occurs.

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.

Note: This includes more than one open 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.

OPNSCOPE

Specifies the extent of influence (scope) of the open operation.

***ACTGRPDFN:** The scope of the open operation is determined by the activation group of the program that called the OVRSAVF command processing program. If the activation group is the

default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

***JOB:** The scope of the open operation is the job in which the open operation occurs.

Example for OVRSAVF

```
OVRSAVF FILE(ONLINE) POSITION(*RRN 100) SECURE(*YES)
```

This command overrides the file named ONLINE so that the first record gotten after the file is opened for input is relative record number 100. The file is also safe from overrides (in previous program calls).

Error messages for OVRSAVF

*ESCAPE Messages

CPF1892

Function &1 not allowed.


OVRTAPF (Override with Tape File) Command Description

OVRTAPF Command syntax diagram

Purpose

The Override with Tape File (OVRTAPF) command is used to (1) override/replace a file named in a program, (2) override certain attributes of a file that is used by a program, or (3) override the file named in a program and override certain attributes of the file processed.

Parameters overridden by this command are specified in the file description, in the program, or in other called file override commands. If a file named in the program is overridden, the name of that file is specified in the FILE parameter and the name of the overriding file is specified in the TOFILE parameter. The OVRTAPF command can also specify parameters to override values contained in the file description of the overriding file. If the file named in the program is not replaced, but certain parameters of the file are overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters overridden are then specified by the other parameters of the OVRTAPF command. Parameters that are not specified do not affect the parameters specified in the file description, in the program, or in other called file override commands.

More information on overriding files is in the File Management topic in the Information Center and the Tape and Diskette Device Programming  book.

Required Parameters

FILE Specifies the name of the file being used by the program to which this override command is applied. If TOFILE(*FILE) is specified, a display device file must be specified. Otherwise, any device file or database file can be specified.

TOFILE

Specifies the qualified name of the tape device file that is used instead of the file specified in the FILE parameter, or if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified on this OVRTAPF command override the other values specified in the tape device file or in the program.

***FILE:** The tape device file named in the FILE parameter has some of its parameters overridden by values specified in this command.

The name of the tape device 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.

tape-device-file-name: Specify the qualified name of the Tape device file that is used instead of the overridden file.

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. The device names in the OVRTAPF command (up to four) override the device names specified in the program or in the tape device 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.

This parameter overrides the volume identifiers specified in the tape device file.

***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.

This parameter overrides the values specified in the device file, in the program, or in other called OVRTAPF commands.

***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.

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.

***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 .

This parameter overrides the data file identifier specified in the device file, in the program, or in other called OVRTAPF commands.

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.

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).

This parameter overrides the value specified in the device file, in the program, or in other called OVRTAPF commands.

***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 RCDBLKFMNT 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. The following *EBCDIC Record Length Values* and *ASCII Record Length Values* tables 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 Values (RCDLEN Parameter)

RCDLKFMT	EBCDIC RCDLEN Ranges	
	FILETYPE(*DATA)	FILETYPE(*SRC)
*F *FB *U	18 - 32767	30 - 32767
*V *VB	1 - 32759	13 - 32767
*VS *VBS	1 - 32759	13 - 32767

RCDLKFMT	ASCII RCDLEN Ranges	
	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 Values (RCDLEN Parameter)

RCDLKFMT	ASCII RCDLEN Ranges	
	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).

This parameter overrides the value specified in the device file, in the program, or in other OVRTAPF commands.

***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 RCDBLKFMNT 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	Maximum
		BLKLEN	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.

This parameter overrides the value specified in the device file, in the program, or in other called OVRTAPF commands.

***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.

RCDBLKFMFMT

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.

This parameter overrides the value specified in the device file, in the program, or in other called OVRTAPF commands.

***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.

***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.

***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. RCDBLKFM(*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.

The following *Required RCDLEN/BLKLEN/BUFOFSET Relation* table 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.

Figure 3. Required RCDLEN/BLKLEN/BUFOFSET Relation

Table 1. Required RCDLEN/BLKLEN/BUFOFSET Relation

CODE	RCDBLKFM T	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.

This parameter overrides the extend value specified in the device file, in the program, or in other called OVRTAPF commands.

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. Records are not added to the end of the specified data

***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

If EXTEND(*YES) is specified, the following values check to see whether the file already exists:

***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 AS/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.

This parameter overrides the value specified in the device file, in the program or in other called OVRTAPF commands.

***DEV D:** 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. >>

Note: For standard labeled (*SL) tapes the CODE parameter is used to determine how the labels are processed. For all label types the TBL, FROMCCSID, and TOCCSID parameters control what conversion, if any, is used for the data portion of the files.



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.

This parameter overrides the value specified in the program, device file, or in other called OVRTAPF commands.

***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.

This parameter overrides the value specified in the program, device file, or in other called OVRTAPF commands.

***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.

Note: Unless an ending option is specified by the high-level language program when the file is closed, this parameter overrides the ending operation specified in the device file, in the program, or in other called OVRTAPF commands.

***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.

» TBL

Specifies the qualified name of a conversion table to be used for single byte conversion of input files or output files. The specified conversion is only used for the data portion of the files. When the specified code is *ASCII (CODE parameter) any labels will be converted between ISO/ASCII 8-Bit code and EBCDIC. When the specified code is *EBCDIC (CODE parameter) the labels, if any, are not converted.

Note: See system supplied conversion tables QSYS/QASCII and QSYS/QEBCDIC for an example of the conversion used to translate between ISO/ASCII 8-Bit code and EBCDIC.

***DFT:** When the specified code is *ASCII (CODE parameter) the data and labels will be converted between ISO/ASCII 8-bit code and EBCDIC. When the specified code is *EBCDIC (CODE parameter) the data and labels will not be converted.

***NONE:** The data will not be converted.

***CCSID:** The CCSID parameters are used to generate a conversion table to use for converting the data portion of the files.

conversion-table-name: Specify the name of a conversion table to use for conversion of the data between single byte character sets.

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

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

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

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

FROMCCSID

Specifies a single byte CCSID used for the input data. The input data is the data read from the tape for input operations, or read from a file for output operations.

from-CCSID: The requested CCSID value is used. The value is validated to ensure that a single byte CCSID is specified. Valid values range from 1 through 65533.

TOCCSID

Specifies the single byte CCSID used for the output data. The output data is the data written to the tape for output operations, or written to a file for input operations.

to-CCSID: The requested CCSID value is used. The value is validated to ensure that a single byte CCSID is specified. Valid values range from 1 through 65533. <<

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.

This parameter overrides the wait time specified in the program or in the device 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 file 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.

SECURE

Specifies whether this file is safe from the effects of previously called file override commands. If SECURE is not specified, processing occurs as if SECURE(*NO) is specified.

***NO:** This file is not protected from the effects of other file overrides; its values can be overridden by the effects of previously called file override commands.

***YES:** This file is protected from the effects of any file override commands previously called.

OVRSCOPE

Specifies the extent of influence (scope) of the override.

***ACTGRPDFN:** The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL:** The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB:** The scope of the override is the job in which the override occurs.

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.

Note: This includes many open files 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.

OPNSCOPE

Specifies the extent of influence (scope) of the open operation.

***ACTGRPDFN:** The scope of the open operation is determined by the activation group of the program that called the OVRTAPF command processing program. If the activation group is the

default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

***JOB:** The scope of the open operation is the job in which the open operation occurs.

Examples for OVRTAPF

Example 1: Overriding a File

```
OVRTAPF FILE(OUT) VOL(DPT706) LABEL(STATUSR)
```

This command overrides a file named OUT in the program using the data file STATUSR on tape volume DPT706.

Example 2: Allowing DBCS Data

```
OVRTAPF FILE(IGCLIB/IGCTAP) IGCDTA(*YES)
```

This command overrides the tape device file named IGCTAP, which is stored in the library IGCLIB, so the file may contain double-byte character set data.

Example 3: Using Data Density of 1600 Bits Per Inch

```
OVRTAPF FILE(OUT) DENSITY(1600)
```

This command overrides a file named OUT to use a data density of 1600 bits per inch when writing to the tape volume.

➤ Example 4: Using a conversion table to process a tape with EBCDIC labels.

```
OVRTAPF FILE(FILE1) REELS(*SL) CODE(*EBCDIC)
        TBL(LIB1/TABLE1)
```

This command overrides a tape device file named FILE1 to specify that a conversion table named LIB1/TABLE1 is to be used to convert all data read from, or written to, the tape volume.

Example 5: Using specified CCSIDs to process a non-labeled tape.

```
OVRTAPF FILE(FILE2) REELS(*NL) TBL(*CCSID)
        FROMCCSID(819) TOCCSID(37)
```

This command overrides a tape device file named FILE2 to specify that any data read from, or written to, the tape volume is to be converted from CCSID 819 to CCSID 37. <<

Error messages for OVRTAPF

*ESCAPE Messages

CPF1892

Function &1 not allowed.



PKGINSOBJ (Package Installable Object) Command Description

Note: To use this command, you must have the 5722-MG1 (Managed System Services for iSeries) licensed program installed.

PKGINSOBJ Command syntax diagram

Purpose

The Package Installable Object (PKGINSOBJ) command saves a copy of one or more objects of any file system and the associated name of the target library, folder or path where they must be created into an installable object. It also creates a distribution catalog entry and loads the installable object that contains the saved objects into the distribution repository.

Restrictions:

1. The GLBNAME consists of up to 9 tokens with the following format:
 - At least 3 tokens should be specified.
 - Only 1 token should contain the REF value and must be any token from the second to the eighth.
 - A refresh level token must follow the REF token. This token must contain only characters from 0-9 and it must be the last token in the global name.
 - The token values MEM, LIB, OBJ, UPD, FIX, or CVRLTR cannot be specified.
2. An installable object that is packaged with objects from the QSYS.LIB file system cannot contain objects from any other file system.
3. An installable object that is packaged with objects from the QDLS file system cannot contain objects from any other file system.
4. This command is shipped with public *EXCLUDE authority
5. The user to run this command must have the authority necessary to perform the SAVxxx commands over the object to be packaged, to the ADDDSTCLGE command, the Delete File (DLTF) command, Create Save File (CRTSAVF) command, and their required authorities.
6. At least one *INCLUDE value must be specified in the OBJ parameter
7. For names involving /QSYS.LIB:
 - a. OBJ must have only one name.
 - b. OBJ(object-path-name) must be one of the following:
 - ('/QSYS.LIB/libname.LIB')
 - ('/QSYS.LIB/libname.LIB/*')
 - ('/QSYS.LIB/libname.LIB/*.type')
 - ('/QSYS.LIB/libname.LIB/objname.type')
 - ('/QSYS.LIB/libname.LIB/objname.FILE/*')
 - ('/QSYS.LIB/libname.LIB/objname.FILE/*.MBR')
 - ('/QSYS.LIB/alib.LIB/anobj.FILE/ambr.MBR')
 - ('/QSYS.LIB/*.type')
 - ('/QSYS.LIB/objname.type')
 - ('/QSYS.LIB/filename.FILE/*')
 - ('/QSYS.LIB/filename.FILE/*.MBR')
 - ('/QSYS.LIB/filename.FILE/membername.MBR')
 - c. .type must be an object type supported by SAVOBJ and RSTOBJ commands
 - d. .libname cannot be QSYS, QDOC..., QTEMP, QSPL, QSRV, QRECOVERY, or QRPLOBJ if libname.LIB is the last component of the name
 - e. OBJ(install-to) must be *SAME or '/QSYS.LIB/libname.LIB'
 - f. SUBTREE must be *ALL
8. For names involving only /QDLS:
 - OBJ must have only one name.
 - OBJ (object-path-name install-to) and SUBTREE must be one of the following:
 - ('/QDLS/path/foldername' '/QDLS/path/foldername') SUBTREE(*ALL)
 - ('/QDLS/path/documentname' '/QDLS/path/documentname') SUBTREE(*OBJ)
9. For names involving other file systems:

- OBJ cannot contain QSYS.LIB or QDLS file systems
10. For names involving links:
- When a link is used to package an object, a link referring to the same object name must exist in the managed system where the installable object is installed. If the link does not exist in the managed system, the user must also package the link.

Required Parameters

GLBNAME

Specifies the token values of the global name. The global name is the name by which the object is known in a system network architecture (SNA) network. The global name can be a maximum of 65-*n* characters in length, where *n* is the number of tokens. A maximum of 9 tokens can be specified and each token can be a maximum of 16 characters in length.

Valid tokens consist of uppercase letters A through Z and numbers 0 through 9. The special characters #, \$, or @ may be used. In multiple-language networks, language translation may make the value not valid when the special characters are used. Use of these characters is not recommended.

Element 1: Token 1

***NETID:** The first global name token value is a network ID generated by the command from the network attributes.

global-name-token-1: Specify the first token of the global name. The first token is recommended to be the registered enterprise ID or network ID.

Element 2-9: Token 2-9

***NETID:** Identifies the global name token *n* value as a network ID. This value is generated from the network attributes.

***CPNAME:** Identifies the global name token value as a control point name. This value is generated from the network attributes.

***DATE:** Identifies the global name token value as the current date. This value is generated from the system value with the format Y1992M04D10.

***TIME:** Identifies the global name token value as the current time. This value is generated from the system value with the format H13M30S20.

global-name-token-n: Specify a token of the global name.

OBJ Specifies the objects to package and where they will be installed. A maximum of 300 object patterns can be specified.

Element 1: Object Name

****:** All objects in the current directory are saved.

object-path-name: Specify an object path name or a pattern that can match many names. The path name can be up to 5000 characters.

Element 2: Include or Omit

The second element specifies whether names that match the pattern should be included or omitted from the operation.

***INCLUDE:** Specify that objects matching the object name pattern are to be packaged, unless overridden by an *OMIT specification.

***OMIT:** Specify that objects matching the object name pattern are not to be packaged. This overrides a *INCLUDE specification as is intended to be used to omit a subset of a previously selected pattern.

Element 3: Install to

Specify where the object will be installed.

***SAME:** Specify that the objects are to be installed with the same names they had when they were packaged. If *OMIT is specified in the second element, this will be ignored.

install-to: Specifies an object path name or a pattern that can match many names where the object will be installed. If a pattern is specified in element 1, the new name must be the directory into which to install any objects that match the pattern. If *OMIT is specified in the second element, this will be ignored. The path name can be up to 5000 characters.

SUBTREE

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

***ALL:** The entire subtree of each directory that matches the object name pattern is included.

***DIR:** Objects in the first level of each directory that matches the object name pattern are included.

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

Note: When *OBJ is specified in the SUBTREE parameter and install-to is *SAME, the object name pattern must exist in the managed system in order to be installed.

TGTRLS

Specifies the release of the operating system on which you intend to use the objects.

***CURRENT:** The objects are to be used on the release of the operating system currently running on your system.

***PRV:** The objects are to be used on the previous release.

release level: Specify the release level in the format VxRxMx. The objects can be used on a system with the specified release or with any later release of the operating system installed. Valid values depend on the current version, release, and modification level, and change with each new release.

AUTL Specifies the name of the authorization list of the objects.

QCQRPSAUTL: The SNA/FS authorization list.

authorization-list-name: The name of the authorization list. The authorization list must already exist.

Examples for PKGINSOBJ

Example 1: Packaging All Objects in Current Directory and Subdirectories

```
PKGINSOBJ  GLBNAME(PACKAGE CURRENT DIRECTORY REF 001)
           OBJ(('*' *INCLUDE *SAME))  SUBTREE(*ALL)
           TGTRLS(*CURRENT)  AUTL(QCQRPSAUTL)
```

This command packages all the objects in the current directory and its subdirectories. When installed, the packaged objects are created in the current directory of the user under which the install request runs. The default OBJ value can be used. The current directory is resolved during the packaging.

Example 2: Packaging All Objects in Current Directory

```

PKGINSOBJ
  GLBNAME(PACKAGE CURRENT DIRECTORY NO SUBDIR REF 002)
  OBJ(('*' *INCLUDE *SAME)) SUBTREE(*OBJ)
  TGTRLS(*CURRENT) AUTL(QCQRPSAUTL)

```

This command packages all the objects in the current directory but not in the subdirectories. The current directory is resolved during the packaging. The objects are installed in the directory specified in the path name that must already exist in the managed system at the time of the installing.

Example 3: Packaging All Objects in Specified Directory

```

PKGINSOBJ
  GLBNAME(PACKAGE A DIRECTORY OMITTING REF 001)
  OBJ('/A' *INCLUDE *SAME)
    ('/A/B/C' *OMIT)) SUBTREE(*ALL)
  TGTRLS(*CURRENT) AUTL(QCQRPSAUTL)

```

This command packages all the objects in directory /A and its subdirectories, except those in directory /A/B/C. If the directory does not already exist, when the objects are installed, the directory /A is created including its subdirectories and their objects.

Example 4: Packaging All Files

```

PKGINSOBJ
  GLBNAME(PACKAGE ALL FILES IN MYLIB REF 003)
  OBJ('/QSYS.LIB/MYLIB.LIB/*.FILE'
    *INCLUDE *SAME))
  SUBTREE(*ALL) TGTRLS(*CURRENT)
  AUTL(QCQRPSAUTL)

```

This command packages all files in MYLIB. When it is installed, the library MYLIB is created, including its objects.

Example 5: Packaging Object in One Library and Installing Object in Another Library

```

PKGINSOBJ
  GLBNAME(RENAMING OBJECTS
    WHEN INSTALLING REF 001)
  OBJ('MyDir/X.PGM'
    *INCLUDE 'YourDir/Y.PGM'))
  SUBTREE(*ALL) TGTRLS(*PRV)
  AUTL(QCQRPSAUTL)

```

This command packages program X from MyDir and installs the object in YourDir with Y name. The system where it will be installed is in the previous release.

Example 6: Packaging Objects in Different File Systems

```

PKGINSOBJ
  GLBNAME(PACKAGE A FILE REF 01)
  OBJ('/MyDir/MyFile'
    *INCLUDE *SAME))
  SUBTREE(*ALL) TGTRLS(*CURRENT)
  AUTL(QCQRPSAUTL)

PKGINSOBJ
  GLBNAME(PACKAGE A DATABASE REF 01)
  OBJ('/QSYS.LIB/MYLIB.LIB/MYFILE.FILE'
    *INCLUDE *SAME))
  SUBTREE(*ALL) TGTRLS(*CURRENT)
  AUTL(QCQRPSAUTL)

```

```

PKGINSOBJ
  GLBNAME(PACKAGE A DOCUMENT REF 03)
  OBJ((' /QDLS/MYFLR/MYDOC'
    *INCLUDE *SAME))
  SUBTREE(*OBJ) TGTRLS(*CURRENT)
  AUTL(QCQRPSAUTL)

```

This command packages a stream file MyFile, a database file MYFILE, and a document MYDOC.

For the examples that follow, the following directory should be taken into account:

```

/A/B
/A/C
/A/D
/A/A1
/A/A1/E
/A/A1/F
/A/A1/G
/A/A1/A2
/A/A1/A2/H
/A/A1/A2/I
/A/A1/A2/J

```

Example 7: Packaging All Objects from Previous Path Name

```

PKGINSOBJ
  GLBNAME(PACKAGE ALL OBJECTS REF 001)
  OBJ((' ../*' *INCLUDE *SAME))
  SUBTREE(*ALL) TGTRLS(*CURRENT)
  AUTL(QCQRPSAUTL)

```

This command packages all the objects from the previous path name.

Example 8: Packaging All Objects in the First Level of Each Directory

```

PKGINSOBJ
  GLBNAME(PACKAGE FIRST LEVEL OBJECTS REF 001)
  OBJ((' /A' *INCLUDE *SAME))
  SUBTREE(*DIR) TGTRLS(*CURRENT)
  AUTL(QCQRPSAUTL)

```

This command packages all the objects in the first level of each directory that matches the object name pattern. The objects that are packaged are:

```

/A/B
/A/C
/A/D
/A/A1
/A/A1/E
/A/A1/F
/A/A1/G
/A/A1/A2

```

Example 9: Packaging Only Objects in the Directory

```

PKGINSOBJ
  GLBNAME(PACKAGE ONLY OBJECTS REF 001)
  OBJ((' /A' *INCLUDE *SAME))
  SUBTREE(*OBJ) TGTRLS(*CURRENT)
  AUTL(QCQRPSAUTL)

```

This command packages only the objects in the directory. The objects that are packaged are:

```

/A/B
/A/C
/A/D
/A/A1

```


Example 10: Packaging All Objects in the User's Home Directory

```
PKGINSOBJ
  GLBNAME(PACKAGE HOME DIR OBJECTS REF 01)
  OBJ(('~' *INCLUDE *SAME))
  SUBTREE(*ALL) TGTRLS(*CURRENT)
  AUTL(QCQRPSAUTL)
```

This command packages all objects from the home directory.

Example 11: Packaging All Objects in the User OTHERUSER Home Directory

```
PKGINSOBJ
  GLBNAME(PACKAGE OTHER HOME DIR REF 01)
  OBJ(('~OTHERUSER' *INCLUDE *SAME))
  SUBTREE(*ALL) TGTRLS(*CURRENT)
  AUTL(QCQRPSAUTL)
```

This command packaged all objects from the OTHERUSER home directory.

Error messages for PKGINSOBJ

*ESCAPE Messages

CPF2105

Object &1 in &2 type *&3 not found.

CPF2110

Library &1 not found.

CPF2283

Authorization list &1 does not exist.

CPF3781

Library &1 not found.

CPF3826

*INCLUDE object required on OBJ parameter.

CPF382C

OBJ parameter value not valid for QSYS file system.

CPF382F

OBJ parameter value not valid for QDLS file system.

CPF5702

File either not DDM file or not found.

CPF9802

Not authorized to object &2 in &3.

CPF9838

User profile storage limit exceeded.

CPF9870

Object &2 type *&5 already exists in library &3.

MSS0114

Not authorized to distribution catalog.

MSS0116

Maximum global name length exceeded.

MSS0117

Global name token &3 not valid. Reason code &4.

- MSS011B**
Distribution catalog entry not found.
- MSS0123**
Internal processing error occurred.
- MSS0124**
Error while managing distribution catalog.
- MSS0125**
Distribution catalog damaged.
- MSS0133**
Not authorized to add distribution catalog entry.
- MSS0136**
Global name already exists.
- MSS02EF**
Not authorized to user profile &1.
- MSS02F0**
User profile &1 not found.
- MSS02F1**
User profile &1 not accessible.
- MSS02F6**
Installable object not packaged.
- MSS02F7**
Global name not valid for installable object.
- MSS02F8**
&1 objects packaged. &2 objects not packaged.
- MSS02F9**
Parameters not valid with multiple file systems.
- MSS02FA**
SUBTREE should be *ALL when QSYS is specified.



PKGPRDDST (Package Product for Distribution) Command Description

Note: To use this command, you must have the 5722-MG1 (Managed System Services for iSeries) licensed program installed.

PKGPRDDST Command syntax diagram

Purpose

The Package Product for Distribution (PKGPRDDST) command saves a copy of the objects that make up a product into a save file so the product can be distributed electronically. A distribution catalog entry is created for the product, and the packaged product is loaded into the distribution repository.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. You must have the authority necessary to perform the Save Licensed Program (SAVLICPGM) command on the product to be packaged to run this command.

3. This command has the same restrictions as the SAVLICPGM command.

Required Parameter

PRDID

Specifies the 7-character identifier of the product to be saved.

Optional Parameters

RLS Specifies which version, release, and modification level of the product is saved.

***ONLY:** Only one version, release, and modification level is installed for the product option.

release-level: Specify the release level in the format VxRxMy, where Vx is the version number, Rx is the release number, and My is the modification number. Valid values for x range from 0 through 9. Valid values for y range from 0 through 9 and A through Z.

OPTION

Specifies which optional parts of the product identified in the Product ID (PRDID) parameter are saved.

***BASE:** Only the base part of the product is saved.

product-option-number: Specify the option number for the product load being saved. Valid values range from 1 through 99.

LODTYPE

Specifies the product load objects being saved.

***ALL:** Code and language objects specified on the LODID parameter are saved.

***CODE:** The objects associated with this product load are saved.

***LNG:** The objects associated with the national language version (NLV) identified on the LODID parameter are saved.

LODID

Specifies the load identifier used for the save operation.

***ALL:** All languages for this product option are saved.

***CODE:** The code load is used.

product-load-ID: Specify the code load to be used. When LODTYPE(*LNG) or LODTYPE(*ALL) is specified, the load ID must be one of the valid IBM national language versions and be specified in the form 29xx.

SAVF Specifies the qualified name of the save file that contains the product packaged for distribution.

***NONE:** A save file containing the product is not provided to package a product for distribution.

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

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

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

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

save-file-name: Specify the name of the save file containing the product packaged for distribution.

AUTL Specifies the name of the authorization list of the distribution repository object. This parameter is valid only when SAVF(*NONE) is specified.

QCQRPSAUTL: The default authorization list is used.

authorization-list-name: Specify the name of an existing authorization list.

REPLACE

Specifies whether the existing packaged product is replaced if the packaged product already exists in the distribution repository.

***NO:** The existing packaged product is not replaced.

***YES:** The existing packaged product is replaced.

TGTRLS

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

***CURRENT:** The object is used on the release of the operating system currently running on your system. If V5R2M0 is running on your system, *CURRENT means that you intend to use the object on a system with V5R2M0 installed. The object can also be used on a system with any later release of the operating system installed.

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

Valid values depend on the current version, release, and modification level, and they change with each new release.

Examples for PKGPRDDST

Example 1: Package the Base Option for Distribution

```
PKGPRDDST PRDID(5722PT1) OPTION(*BASE)
```

This command saves the BASE option of the Performance Tools licensed program for both the code and language parts. It then creates the following distribution catalog entry and stores the saved product into the distribution repository:

```
I3IBM1 AS400 5722PT1 V5R2M0 BASE ALL ALL REF 001 V5R2M0
```

Example 2: Package the Program Objects

```
PKGPRDDST PRDID(ACCOUNT) RLS(V5R2M0)  
LODTYPE(*CODE) LODID(*CODE)
```

This command packages the V5R2M0 ACCOUNT product for distribution and saves only the program objects for the product. The command also creates the following distribution catalog entry and stores the saved product into the distribution repository:

```
I3IBM1 AS400 ACCOUNT V5R2M0 BASE CODE CODE REF 001 V5R2M0
```

Example 3: Package the Language Objects

```
PKGPRDDST PRDID(ACCOUNT) LODTYPE(*LNG) LODID(2924)
```

This command packages the English version of the ACCOUNT product for distribution and creates the following distribution catalog entry and stores the saved product into the distribution repository:

```
I3IBM1 AS400 ACCOUNT V5R2M0 BASE LNG 2924 REF 001 V5R2M0
```

Example 4: Package the Product for Distribution

```
PKGPRDDST PRDID(BILLS01) SAVF(*LIBL/BILLSAVF)
```

This command packages the V5R2M0 BILLS01 product for distribution for both the code and language parts. The product is not saved because the save file containing the product was specified. This command also creates the following distribution catalog entry and stores the saved product into the distribution repository:

```
I3IBM1 AS400 ACCOUNT V5R2M0 BASE ALL ALL REF 001 V5R2M0
```

Error messages for PKGPRDDST

***ESCAPE Messages**

CPF3D94

No product found in save file.

CPF37xx

Save/restore error messages.

CPF3805

Objects from save file &1 in &2 not restored.

CPF3812

Save file &1 in &2 in use.

CPF81xx

Damaged object error messages.

CPF98xx

Common error messages.

MSS0123

Internal processing error occurred.

MSS0144

Distribution catalog entry not retrieved.

MSS020C

Product and save file information do not match.

MSS020D

Required object not found or damaged.

MSS020F

Required object locked.

MSS0210

Not authorized to perform operation.

MSS0211

Product already packaged for distribution.

MSS0212

Product could not be packaged for distribution.

MSS022A

Object &2/&1 not found.

MSS022C

Cannot specify QTEMP for save file library.

MSS0415

Managed system attributes not found or damaged.



PKGPRDOPT (Package Product Option) Command Description

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

PKGPRDOPT Command syntax diagram

Purpose

The Package Product Option (PKGPRDOPT) command packages one or more product loads for a specified product option enabling the loads to be saved using the Save License Program (SAVLICPGM) command.

The following notes provide information on how the command works.

Notes:

1. To package a product load that has folders, you must be enrolled in the system distribution directory and you must have *ALL authority to the folders.
If you have authority for this command, you can package any product created using the System Manager licensed program regardless of whether you have authority to the objects for the product.
2. The only object types that the PKGPRDOPT command will package are those listed in the Save Object (SAVOBJ) command. Other object types are ignored.

Required Parameters

PRDID

Specifies the product identifier (ID) of the product option being packaged. The ID must be seven characters in length.

OPTION

Specifies the product option being packaged.

***BASE:** The base option is packaged.

product-option-number: Specify the option number of the product to be packaged. Valid values range from 1 through 99.

ALWAPICHG

Specifies whether to prevent changes to the description of each object captured by this command.

***SAME:** The allow change by program attribute of the objects being packaged is not changed by this command.

***NO:** The object description for each object being packaged cannot be changed by the QLICOBJD application program interface (API) or the Change Product Object Description (CHGPRDOBJD) command.

Optional Parameters

LODID

Specifies the load ID of the product loads being packaged.

***ALL:** All product loads for which a product load object exists are packaged.

If only a code load exists, the code load is packaged.

***CODEDFT:** The default code load, 5001, is packaged.

product-load-ID: Specify a language load ID or the default code load, 5001, for a code load.

RLS Specifies the version, release, and modification level of the product option.

***ONLY:** The release level is determined by searching the system for a product definition for the given product ID. The release level is taken from the product definition. This value is not valid if more than one product definition exists for the same product ID.

version-release-modification: Specify the version, release, and modification level of the product being packaged.

REPACKAGE

Specifies whether to allow the command to package a product option which has already been packaged.

***NO:** Product options which have been packaged are not repackaged.

***YES:** Packages all specified product loads regardless of their current state.

Example for PKGPRDOPT

```
PKGPRDOPT PRDID(9XYZ123) OPTION(*BASE) ALWAPICHG(*SAME)
```

This command packages the base option of the product 9XYZ123.

Error messages for PKGPRDOPT

***ESCAPE Messages**

CPF0CB2

Product identifier &1 not valid.

CPF0CEB

Product loads not packaged.

CPF0CE2

&9 product loads packaged, &10 product loads not packaged.

CPF0CE3

&9 product loads packaged, &10 product loads not packaged. See job log.

CPF0CE7

Product &1 release &2 not packaged.

CPF0CFC

Product definition not found.

CPF0CFD

Code load not found for product &1 release &2 option &3.

CPF0CFF

Multiple releases available.

CPF0C4B

Product availability object &2/&1 recovery required.

CPF0C4C

Cannot allocate object &1 in library &2.

CPF0C4D

Error occurred while processing object &1 in library &2.

CPF8122

&8 damage on library &4.

CPF0CB2

Product identifier &1 not valid.

CPF8191

Product definition &4 in &9 damaged.

CPF8193

Product load object &4 in &9 damaged.

CPF9012

Start of document interchange session not successful for &1.

CPF9032

Document interchange session not started.

CPF9803

Cannot allocate object &2 in library &3.

CPF9830

Cannot assign library &1.



PING Command

PING Command
For the description of the PING command, see the VFYTCPCNN (Verify TCP/IP Connection) command description.

POSDBF (Position Database File) Command Description

POSDBF Command syntax diagram

Purpose

The Position Database File (POSDBF) command allows the user to set the position of a database file to either the beginning or end of an open file.

Required Parameters

OPNID

Specifies which opened file changes position. This file must be opened by either the Open Database File (OPNDBF) or Open Query File (OPNQRYF) command.

POSITION

Specifies the starting or ending position of the database file.

***START:** The position of the database file is set to the start position of the member currently open. After the start position is set, a read next operation gets the first record in the current member. If the Override Database File (OVRDBF) command MBR(*ALL) processing is in effect, a read previous operation gets the last record in the previous member. If a read previous is done and the previous member does not exist, the end of the file message (CPF5001) is sent.

***END:** The position of the database file is set to the end of the member that is currently open. After the end position is set, a read previous operation gets the last record in the current member. If the Override Database File (OVRDBF) command MBR(*ALL) processing is in effect, a read next operation gets the first record in the next member. If a read next is done and the next member does not exist, the end of the file message (CPF5001) is sent.

Example for POSDBF

```
POSDBF OPNID(XXX) POSITION(*START)
```

This command sets the record position of the database file that is opened with OPNID(XXX) to the starting position of the database file member that is currently open.

Error messages for POSDBF

*ESCAPE Messages

CPF5213

Positioning of member &3 failed.

CPF5230

No file open with OPNID(&4).

PWRDWNSYS (Power Down System) Command Description

PWRDWNSYS Command syntax diagram

Purpose

The Power Down System (PWRDWNSYS) command prepares the system for ending and then starts the power-down sequence. All active subsystems are notified that the system is being powered down; no new jobs or routing steps can be started by any subsystem. For example, jobs that are on a job queue as a result of a Transfer Job (TFRJOB) command are not allowed to complete. During the subsequent initial program load (IPL), they are removed from the job queue and their job logs are produced. When the system is powered down with the *CNTRLD option, a vary off of configuration objects is initiated, but may not complete before the power down completes. When the system is powered down with the *IMMED option, no vary off of configuration objects is performed.

Note:

If network server descriptions are configured on the system, all NWSDs should be varied off **before** the Power Down System command is issued to ensure the integrity of system and user data associated with each network server.

Note:

If tape units are installed on the system, all tape reels that are on the devices should be unloaded **before** the system is powered down. This step ensures the integrity of data on the tapes.

Note:

On a partitioned system, powering down the primary partition will cause the other partitions to power down. Ensure the other partitions are ready to be powered down before powering down the primary partition.



Note:

If independent auxiliary storage pool (ASP) devices are configured on the system, all independent ASPs should be varied off before the Power Down System command is issued. This allows the work associated with each independent ASP to end in an orderly manner, which helps ensure the consistency of data associated with each independent ASP.

Note:

The Power Down System exit point (QIBM_QWC_PWRDWNSYS) can be used to register a program that is called when the PWRDWNSYS command is used. This program can perform clean up functions before the system is powered down.



Restrictions: To run this command, the user must have job control (*JOBCTL) authority. » The following restrictions apply:

1. You must have *USE authority to the image catalog specified by the IMGCLG parameter and *EXECUTE authority to the QUSRSYS library containing that image catalog.



Optional Parameters

OPTION

Specifies whether the system either allows the active subsystem to end processing of active jobs in a controlled manner (which lets the application program perform end processing) or the system ends the jobs immediately. In either case, the system performs certain job-cleanup functions.

***CNTRLD:** The subsystem, within the time specified by the DELAY parameter, ends all active jobs in a controlled manner. During that time, programs running in those jobs are allowed to perform job cleanup (end-of-job processing) functions. If an active job could begin to loop or send an inquiry message to QSYSOPR, an appropriate time delay should be specified by using the DELAY parameter.

***IMMED:** The subsystem ends all active jobs immediately. This means the programs running in those jobs are not allowed to perform any job cleanup. Thus, a minimum amount of time is required when *IMMED is specified. The amount of time allowed for cleanup when *IMMED is specified is controlled by QPWRDWNLMT, the system value. This option might cause undesirable results if data has been partially updated, and it should be used only after a controlled end has been unsuccessfully attempted.

When OPTION(*IMMED) is specified while the system is operating under auxiliary power, or if the delay time specified in the DELAY parameter ends while the system is under auxiliary power, the system ignores the QPWRDWNLMT system value and starts the power-down sequence without additional job cleanup activity.

DELAY

Specifies the number of seconds that the system allows a controlled end of processing operation to be performed by the active subsystems. If the end-of-job cleanup functions are not finished within the specified delay time, any remaining jobs are ended immediately.

3600: The amount of time in which to complete a controlled end of processing is limited to 3600 seconds.

***NOLIMIT:** The system does not power down until the last job is complete.

Note:

If *NOLIMIT is specified and a batch job begins to loop, the system does not power down.

delay-time: Specify the maximum number of seconds in which a controlled end operation can be performed. Valid values range from 1 through 99999 seconds.


RESTART

Specifies whether the system ends and powers down, or ends and then restarts in unattended mode.

***NO:** The system ends and powers down.

Element 1: Restart after power down

***YES:** If the system is on utility power, it undergoes end of system processing (but does not power down) and then does an initial program load (IPL). If the system is on auxiliary power, it powers down and does an automatic IPL when utility power is restored (if the QPWRRSTIPL system value is set to '1'). When the system restarts or an automatic IPL occurs, the IPL proceeds in an unattended mode. In unattended mode, displays such as the *IPL options* prompt are not shown.

More information is in the Work Management  book.

Element 2: Restart type

Specifies the point from which the initial program load (IPL) restarts. Specifying *SYS rather than *FULL can reduce the time required to restart the system.

***IPLA:** The value specified on the Change IPL Attributes (CHGIPLA) command is used. To determine the current setting for this value, use the Display IPL Attributes (DSPIPLA) command.

***SYS:** The operating system is restarted. The hardware will only be restarted when required by the system.

***FULL:** All portions of the system, including the hardware, are restarted.

IPLSRC

Specifies whether an initial program load (IPL) is started from the A-source, B-source, or D-source of the system. This parameter allows the user to control which Licensed Internal Code (LIC) storage source of the system to use at IPL. Also, the source of the system determines where LIC program temporary fixes (PTFs) are applied. ➤ This parameter also allows the system to be upgraded to a new release from an install image on DASD. ⏪

Source Considerations:

LIC has three storage areas, known as the A-source, the B-source, and the D-source. The D-source is a tape device or optical device. The A- and B-sources are part of the system memory. Initially, the A- and B-sources are identical, but when Licensed Internal Code fixes (PTFs) are applied temporarily, the temporary fixes are stored only on the B-source. When these fixes become permanent, they are copied from the B-source to the A-source; therefore, the fixes reside on both the A-source and B-source.

The user who wants to send temporary fixes to the B-source must start the system from the A-source, which causes the fixes to be sent to the opposite source, or the B-source.

A user that starts the system from the A-source is running the system from the permanent fixes. A user that starts the system from the B-source is running the system from a combination of temporary and permanent fixes. A user that starts the system from the D-source uses the Licensed Internal Code loaded from the media.

Notes:

1. It is recommended that the user specify RESTART(*YES), otherwise, the user cannot be assured as to which source the system is actually starting. This precaution can save the user time.

***PANEL:** The system is started from the source (A-source, B-source, or D-source) that is currently shown on the operator's display,

A: The system is started from the A-source.

B: The system is started from the B-source.

D: The system is started from the D-source, the install media.

➤ ***IMGCLG:** The system is started from the image catalog specified with the IMGCLG parameter. RESTART(*YES) must be used when this option is selected.

IMGCLG

Specifies the image catalog used when IPLSRC(*IMGCLG) is selected. After the system is powered down, an install using the specified image catalog is performed. See the Work with Image Catalog (WRKIMGCLG) command for more information. RESTART(*YES) must be used when this parameter is specified.

image-catalog-name: Specify the name of the image catalog in library QUSRSYS. <<

ENDSBSOPT

Specifies the options to take when ending the active subsystems. In general, specifying these options will improve the performance of the PWRDWNSYS command. Each option has certain side effects that you need to analyze before using that option.

This parameter has no effect on jobs that are already in the ending status.

***DFT:** The subsystems will end with no special ending options.

- Joblogs will be produced.
- The run priority will not change.
- The timeslice value will not change.

***NOJOBLOG:** No joblogs will be created for jobs that are ended due to this command being invoked. This includes subsystem monitor jobs and all user jobs in the subsystem. This option can significantly reduce the amount of time necessary to complete the PWRDWNSYS command. However, if a problem occurs in a job, there will be no joblog to record the problem, which may make problem diagnosis difficult or impossible.

Note:

If OPTION(*IMMED) is specified, then no joblogs are produced during PWRDWNSYS regardless of the ENDSBSOPT parameter. However, these joblogs will still be produced on the next IPL of the system unless the *NOJOBLOG option is specified. Therefore, if you specify OPTION(*IMMED) ENDSBSOPT(*NOJOBLOG), the system will not power down more quickly, but the subsequent IPL may be faster.

***CHGPTY:** The CPU priority of jobs that are ending is changed to a higher value (worse priority). The remaining active jobs on the system may have better performance when *CHGPTY is specified. However, jobs that are ending may take longer to finish. This option is ignored if the system is ending controlled. But if the DELAY time limit expires, this option will take affect immediately.

***CHGTSL:** The timeslice of jobs that are ending is changed to a lower value. The remaining active jobs on the system may have better performance when *CHGTSL is specified. However, jobs that are ending may take longer to finish. This option is ignored if the system is ending controlled. But if the DELAY time limit expires, this option will take affect immediately.

TIMOUTOPT

Specifies the option to take when the system does not end within the time limit specified by the QPWRDWNLMT system value. If this time limit is exceeded, the subsequent IPL will be abnormal regardless of the value specified for this parameter.

***CONTINUE:** The system will ignore the timeout condition and continue powering the system down. If RESTART(*YES) is specified, the system will restart automatically. A minimum of information will be available for service to debug the system.

***MSD:** The system will issue a main store dump which can be used by service to debug the system. If the main store dump manager is configured correctly, the system will restart after the dump is finished. See the *iSeries Licensed Internal Code Diagnostic Aids - Vol 1* book for more information on main store dump manager.

***SYSREFCDE:** The system will display system reference code B900 3F10 and the system will stop. >>

CONFIRM

Specifies whether the request should be confirmed before the system is powered down.

***ENVVAR:** The value in environment variable QIBM_PWRDWNSYS_CONFIRM is used to determine whether the request should be confirmed. If the value is set to *INTERACT, *YES, or *NO, the action described below for that value is taken. If the environment variable is not defined or not set to one of these values, then there is no confirmation. System initiated power downs do not use the environment variable.

***INTERACT:** A confirmation panel is displayed when the PWRDWNSYS command is issued in an interactive job. There is no confirmation when the PWRDWNSYS command is issued in a non-interactive job.

***YES:** A confirmation panel is displayed when the PWRDWNSYS command is issued in an interactive job. An inquiry message is sent to QSYSOPR when the PWRDWNSYS command is issued in a non-interactive job.

***NO:** There is no confirmation when the PWRDWNSYS command is issued. <<

Examples for PWRDWNSYS

Example 1: Performing An Immediate End

```
PWRDWNSYS OPTION(*IMMED)
```

This command causes the system to perform an immediate end without allowing any active jobs to perform cleanup routines. Once the system completes its end functions, it starts the power-down sequence.

Example 2: Specifying a Controlled End

```
SBMJOB JOB(LASTJOB) JOBD(QBATCH) JOBPTY(9)  
JOBQ(QBATCH) RQSDTA('PWRDWNSYS *CNTRLD 3600')
```

This command submits a low priority batch job that, when run, causes the system to perform a controlled end. The controlled end is allowed one hour (3600 seconds) for completion before any remaining jobs are ended. This method of issuing the PWRDWNSYS command could be used to allow other higher priority jobs on job queue QBATCH (including those that are on the queue as a result of the Transfer Job (TFRJOB) command) to be completed before the PWRDWNSYS command is run. There must be an active subsystem for which the QBATCH job queue is a source of work.

Example 3: Specifying a Controlled End With No Time Limit

```
PWRDWNSYS OPTION(*CNTRLD) RESTART(*YES)
```

This command causes the system to perform a controlled end with no time limit. When all jobs in the system have completed, the system prepares for ending and starts an IPL.

After PWRDWNSYS OPTION(*CNTRLD) is entered, and before the delay time ends, this command can be overridden by entering PWRDWNSYS OPTION(*IMMED). In this case, the values specified or defaulted for the RESTART parameter on the second command also override the values specified or defaulted for the first command.

Example 4: Changing the IPL Source After Immediate End

```
PWRDWNSYS OPTION(*IMMED) RESTART(*YES) IPLSRC(A)
```

This command causes the system to end immediately and change the IPL source to A. When the system restarts, it IPLs on the A source.

Example 5: Allowing the operating system to determine the restart point

```
PWRDWNSYS OPTION(*IMMED) RESTART((*YES *SYS))
```

This command causes the IPL to restart at the point determined by the operating system.

Example 6: Changing the time out option.

```
PWRDWNSYS OPTION(*IMMED) TIMOUTOPT(*MSD)
```

This command causes the system to end immediately. If the QPWRDWNLMT system value is exceeded, the system will dump the main storage. If the main store dump manager is configured correctly, the system will restart. Otherwise, the B900 3F10 system reference code will be displayed and the system will halt.

» Example 7: Installing a new release of the operating system.

```
PWRDWNSYS RESTART(*YES) IPLSRC(*IMGCLG) IMGCLG(MYCAT1)
```

This command causes the system to end and then start installing a new release of the operating system from the image catalog MYCAT1. <<

Error messages for PWRDWNSYS

*ESCAPE Messages

CPF1001

Wait time expired for system response.

CPF1036

System powering down with *CNTRLD option.

CPF1037

System powering down with *IMMED option.

CPF1038

No authority to use command.

CPF1091

Function check occurred in system arbiter. »

CPFBC42

Verify for image catalog &1 failed. <<

PRTACTRPT (Print Activity Report) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

PRTACTRPT Command syntax diagram

Purpose

The Print Activity Report (PRTACTRPT) command generates reports based on the data collected by the Work with System Activity (WRKSYSACT) command.

Optional Parameters

MBR Specifies the member in which the performance data is saved by the Work with System Activity (WRKSYSACT) command.

QAITMON: The standard member name, QAITMON, is used.

member-name: Specify the name of the member that contains the performance data.

LIB Specifies the name of the library where the performance data file, QAITMON, is stored.

QPFRDATA: The performance data is stored in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library where the data file is stored.

TITLE Specifies a title that is printed at the top of each page of the report.

***BLANK**: No title is printed on the activity report.

'report-title': Specify a title of up to 50 characters of text, enclosed in apostrophes.

RPTTYPE

Specifies the type of activity report that is generated.

***SUMMARY**: The top ten entries, as measured over the entire time frame specified by the PERIOD parameter, are listed according to CPU utilization and number of I/O operations performed.

***DETAIL**: For each interval specified by the PERIOD parameter, the number of entries specified by the NBRJOBS parameter are listed in the order specified by the SEQ parameter.

***ALL**: Generates both the summary activity report and the detailed activity report.

PERIOD

Specifies the period of time on which to report. The following values can be coded in this parameter, which contains two lists of two elements each:

```
PERIOD((start-time start-date)
        (end-time end-date))
```

If PERIOD is not specified, the following values are assumed:

```
PERIOD((*FIRST *FIRST) (*LAST *LAST))
```

*N may be used in place of an element that precedes the values that are specified to maintain the position in the parameter value sequence. For example, PERIOD(*N(*N 091289)) specifies the ending date and uses the defaults for the other values.

Element 1: Starting Time

One of the following values is used to specify the starting time. Data collected before the specified time (and date) is not shown.

***FIRST**: Data records starting from the beginning of the start day (00:00:00) are included in the report.

start-time: Specify the starting time (for the specified starting date) after which the data must have been collected in order to be included in the report. 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.

The time is in 24-hour format; for example, use '13:00' for 1 o'clock in the afternoon.

Element 2: Starting Date

One of the following values is used to specify the starting date. Data collected before the specified date (and time) is not included in the report.

***FIRST:** Data records starting from the first day of the collection period are included in the report.

start-date: Specify the starting date after which the data is collected to be included in the report. The date must be entered in the format specified by the system value, QDATFMT, and, if separators are used, using the format specified by the system value, QDATSEP. For example, the system might have a date format of 'mm/dd/yy' where mm=month, dd=day, and yy=year and are all required 1- or 2-digit values. The slashes (/) are optional if all 6 digits are specified. If the slashes are not used, or if the value is entered from the prompt screen, the apostrophes are not required.

Element 3: Ending Time

One of the following values is used to specify the ending time. Data collected after this time (and ending date) is not included in the report.

***LAST:** Data records through the end of the day (23:59:59) are included.

end-time: Specify the ending time (for the specified ending date) before which the data must have been collected to be included in the report. See *start-time* in this parameter for details on how the time must be specified.

Element 4: Ending Date

One of the following values is used to specify the ending date. Data collected after the specified date (and time) is not included in the report.

***LAST:** Data records collected up to the last day (and time) are included in the report.

end-date: Specify the ending date before which the data must be collected to be included in the report. See *start-date* in this parameter for details on how the date must be specified.

SEQ Specifies the field by which the jobs and tasks are ranked and listed on the detailed activity report. This parameter is valid only when *DETAIL or *ALL is specified for the RPTTYPE parameter.

***CPU:** The entries are listed in descending order according to CPU utilization.

***JOBTASK:** The entries are listed alphabetically according to the job or task name.

***USER:** The entries are listed alphabetically according to the user profile.

***PTY:** The entries are listed in descending order according to priority.

***TOTALIO:** The entries are listed in descending order according to the total number of synchronous and asynchronous I/O operations initiated.

***SYNCIO:** The entries are listed in descending order according to total number of synchronous I/O operations performed.

***ASYNCIO:** The entries are listed in descending order according to the total number of asynchronous I/O operations initiated.

***FAULT:** The entries are listed in descending order according to the number of process access group (PAG) faults which occurred.

***SDBREAD:** The entries are listed in descending order according to the number of synchronous database read operations performed.

***SDBWRITE:** The entries are listed in descending order according to the number of synchronous database write operations performed.

***SNDBREAD:** The entries are listed in descending order according to the number of synchronous nondatabase read operations performed.

***SNDBWRITE:** The entries are listed in descending order according to the number of synchronous nondatabase write operations performed.

***ADBREAD:** The entries are listed in descending order according to the number of asynchronous database read operations initiated.

***ADBWRITE:** The entries are listed in descending order according to the number of asynchronous database write operations initiated.

***ANDBREAD:** The entries are listed in descending order according to the number of asynchronous nondatabase read operations initiated.

***ANDBWRITE:** The entries are listed in descending order according to the number of asynchronous nondatabase write operations initiated.

NBRJOBS

Specifies the number of entries listed for each interval in the detailed activity report. This parameter is valid only when *DETAIL or *ALL is specified for the RPTTYPE parameter.

10: Ten entries for each interval are listed.

***ALL:** All the entries contained in the measured data are listed.

number-of-jobs: Specify the number of entries listed for each interval.

JOB Specifies the job name used if the job is submitted for batch processing.

Note: If *NONE is specified on the JOB parameter, this parameter is ignored; job processing is performed interactively.

PRTRPT: The command name is used for the job name.

***MBR:** The name selected for the performance data member on the MBR parameter is used.

job-name: Specify the name used for batch jobs.

JOB 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 to be searched.

QPFRJOB: The IBM-supplied Performance Tools job description, QPFRJOB, is used.

job-description-name: Specify the name of an alternate job description.

Other Single Values

***NONE:** A batch job is not submitted; processing continues interactively while the user waits. The user's work station is not available for other use during this time, which could be significant for long jobs.

Examples for PRTACTRPT

Example 1: Generating a Summary Report

```
PRTACTRPT
```

This command submits a batch job that generates a summary activity report using the performance data found in the default member QAITMON located in the default library QPFRDATA. The report covers the entire measurement period, and the title of the report is left blank.

Example 2: Generating a Summary and Detailed Activity Report

```
PRTACTRPT MBR(JUNE01)
          TITLE('Activity Report for June 1st')
          RPTTYPE(*ALL) SEQ(*CPU)
```

This command submits a batch job that generates both a summary and a detailed activity report. The performance data comes from member JUNE01 located in the default library QPFRDATA. The report covers the entire measurement period, and the title of the report is 'Activity Report for June 1st.' The detailed activity report lists ten entries in descending order according to CPU utilization for each interval.

Error messages for PRTACTRPT

*ESCAPE Messages

PFR7010

No data in member to print.

PFR7017

Cannot print activity report.

PRTADPOBJ (Print Adopting Objects) Command Description

PRTADPOBJ Command syntax diagram

Purpose

The Print Adopting Objects (PRTADPOBJ) command allows you to print a report of the objects that adopt the special and private authorities of the specified user profile. This is a way to check for security exposures associated with program adoption.

This command prints two reports for a user profile. The first report (Full Report) contains all of the objects that adopt the authorities of the user profile. The second report (Changed Report) contains the objects that now adopt the authorities of the user profile that did not adopt the authorities of the user profile when the PRTADPOBJ command was previously run for the user profile. If the PRTADPOBJ command was not previously run for the user profile, there will be no 'Changed Report'. If the command has been previously run for the user profile but no additional objects adopt the authorities of the user profile, then the 'Changed Report' is printed but there are no objects listed.

An object is not listed in the Changed Report when the object itself is changed. Therefore, you should periodically review the entire list of objects that adopt to understand their current function.

Restrictions:

1. You must have *ALLOBJ special authority to use this command.

- The user profile specified on the command is locked while the command is running. The lock prevents such things as objects having their owner changed to this profile. If this profile owns a lot of objects, the profile could be locked for an extended period of time.

Required Parameter

USRPRF

Specifies the name of the user profile whose adopted object information will be printed.

***ALL:** The adopted information is printed for all user profiles.

user-profile-name: Specify the name of the user profile to print the adopted information for.

generic-user-name:* Specify the generic name of the user profile to print the adopted information for. A generic name is a character string of one or more characters followed by an asterisk (*).

Optional Parameter

CHGRPTONLY

Specifies whether just the changed report should be printed.

***NO:** The full and changed reports will be printed.

***YES:** Only the changed report will be printed.

Example for PRTADPOBJ

```
PRTADPOBJ USRPRF(OURSECOFR)
```

This command will produce the full and changed reports for the objects that adopt the special and private authorities of the user profile OURSECOFR.

Error messages for PRTADPOBJ

*ESCAPE Messages

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

PRTAFPDTA (Print Advanced Function Printer Data) Command Description

PRTAFPDTA Command syntax diagram

Purpose

The Print Advanced Function Printer Data (PRTAFPDTA) command prints output received from a System/370* host. This command provides the user with a means of specifying the file being printed and the parameters used to control the print operation.

Required Parameter

FILE Specifies the qualified name of the Advanced Function Printing Data Stream (AFPDS) file to be printed. Only physical files are supported for this command. If you use the Override with Printer File (OVRPRTF) command with PRTAFPDTA, do not override the device type (DEVTYPE parameter).

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 AFPDS to be printed.

Optional Parameters

MBR Specifies the member that contains the data to be printed.

***FIRST:** The first member in the database file is used.

member-name: Specify the member that contains the data to be printed.

DEV Specifies the printer that prints the file.

***JOB:** The printer device specified in the job description is used.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

printer-device-name: Specify the name of the printer.

FORMDF

Specifies the form definition to use when printing the file. A form definition is a resource object that defines the characteristics of the form, including: overlays, position of page data on the form, and number of copies of pages and modification to pages. The form definition is located inline with the file being printed, or in a library.

***DEVD:** The device description obtains the name of the form definition being used. If no value is specified, *DEVD is assumed.

***INLINE:** The form definition that is inline with the file that is printed is used.

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.

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

form-definition-name: Specify the name of the form definition that must exist in the library named. A valid name has up to 8 characters.

COPIES

Specifies, for spooled files, the number of copies being printed.

1: One copy of the output is printed.

number-of-copies: Specify the number of copies that are printed.

STRPAGE

Specifies the page on which printing starts. This parameter is used for partial printing of a file.

1: Printing starts with page 1. If the start page is not specified, 1 is assumed.

starting-page-number: Specify the page number on which printing starts.

ENDPAGE

Specifies the page on which printing ends. This parameter is used for partial printing of a file ending at a specified page number. If both start page and end page are specified, the end page must be greater than or equal to the start page. Specifying an end page beyond the end of the actual file does not create an error condition.

***END:** Printing concludes at the end of the file.

ending-page-number: Specify the page number on which printing ends.

FIDELITY

Specifies the degree of exactness required when printing the file.

***ABSOLUTE:** The job is printed only if the file can be printed exactly as specified by the data stream and external controls.

***CONTENT:** Prints the file using all available exception handling.

Examples for PRTAFPD TA

Example 1: Printing Specific Pages

```
PRTAFPD TA FILE(MYLIB/MYFILE)
STRPAGE(2) ENDPAGE(6)
```

This command prints the first member in file MYFILE in library MYLIB starting with page 2 and ending on page 6.

Example 2: Printing Using All Available Exception Handling

```
PRTAFPD TA FILE(MYLIB/MYFILE)
FORMDF(F10101) FIDELITY(*CONTENT)
```

This command prints the first member in file MYFILE in library MYLIB using a form definition of F10101 and all available exception handling.

Error messages for PRTAFPD TA

*ESCAPE Messages

CPF511B

Data stream not correct for record &2 in file &1.

PQT4001

Data stream not valid in structured field &2 in file &1.

PQT4003

Form definition &2 not found in library.

PQT4004

Starting page number &1 greater than ending page number &2.

PQT4006

Unable to process file &1 because of variable length fields.

PQT4007

Data stream not valid in file &1.

PRTCMDUSG (Print Command Usage) Command Description

PRTCMDUSG Command syntax diagram

Purpose

The Print Command Usage (PRTCMDUSG) command creates a cross-referenced list of a specified group of CL commands that are used in a specified group of CL programs. The report shows, program by program, which of the specified commands are used in each program. The report can be used to identify which programs need to be recompiled because of changes that have been made to the command definition objects of commands specified on the PRTCMDUSG command. Note that this command can take a long time to run and can make a lot of printed output.

Required Parameter

CMD Specifies the qualified names of up to 50 CL commands for which specified programs are searched and printed in a command usage report. The system searches the specified programs for every occurrence of each library-name/command-name character string you specify.

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.

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

command-name: Specify the names of the commands for which specified programs are searched.

Optional Parameter

PGM Specifies the qualified name of the program or the generic name of several programs that are searched for the specified commands. Only the programs and libraries for which the user has some (any) authority are included in the report. This parameter also can specify that all (*ALL) programs in the specified library or libraries (*USRLIBL/*ALLUSR, for example) are searched.

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

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

***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.

>> *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	>> 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 QUSRvRxMx, 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 CL programs in the specified library (or all libraries identified in the library qualifier) for which the user has some authority are searched.

program-name: Specify the name of the program being searched for the specified command.

generic-program-name*: Specify the name of the program or the generic name of several programs in the specified library qualifier that are searched for the specified commands. 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 PRTCMDUSG

```
PRTCMDUSG  CMD(CPYF)  PGM(PAYROLL/*ALL)
```

This commands searches all CL programs in the library PAYROLL for the Copy File (CPYF) commands and prints the names of both the commands and the program.

Error messages for PRTCMDUSG

*ESCAPE Messages

CPF0593

PRTCMDUSG command ended by controlled end.

CPF0595

PRTCMDUSG command ended.

CPF0596

PRTCMDUSG command ended. Cannot open print file.

PRTCMNSEC (Print Communications Security) Command Description

PRTCMNSEC Command syntax diagram

Purpose

The Print Communications Security (PRTCMNSEC) command allows you to print a report containing the security attributes of the *DEV, *CTLD and *LIND objects currently on the iSeries 400. This command provides a way to check the security of your communications configuration on the iSeries 400.

The Print Communications Security command creates two spooled output files containing communications security information. The first spooled output file contains a report generated by the Display Configuration List (DSPCFGL) CL command. This report contains the entries currently in the APPN remote configuration list QAPPNRMT. If the QAPPNRMT configuration list does not exist on the system then no report is printed. The second spooled output file contains the security attributes of the *DEV, *CTLD and *LIND objects on the system.

The spooled output file containing the *DEV, *CTLD and *LIND objects contains two reports. The first report (Full Report) contains all of the communications objects and prints the security attributes of each object. The second report (Changed Report) contains the communications objects that have changed since the PRTCMNSEC command was last run. If the PRTCMNSEC command was not previously run, there will be no 'Changed Report'. If the command has been previously run but no communication object information has changed then the 'Changed Report' is printed but there are no objects listed.

Restrictions: You must have *ALLOBJ and *IOSYSCFG special authority to use this command.

Optional Parameter

CHGRPTONLY

Specifies whether just the changed report should be printed.

***NO:** The full and changed reports are printed.

***YES:** Only the changed report is printed.

Example for PRTCMNSEC

```
PRTCMNSEC
```

Both the full and change report will be generated for the communication security information.

Error messages for PRTCMNSEC

*ESCAPE Messages

CPFB307

Command &1 in use in another job.

PRTCMNTRC (Print Communications Trace) Command Description

PRTCMNTRC Command syntax diagram

Purpose

The Print Communications Trace (PRTCMNTRC) command transfers the communications trace data for the specified line, network interface description, or network server description to a spooled file or an output file.

» The PRTCMNTRC command can also be used to format communications trace data that was previously dumped to a stream file using the Dump Communications Trace (DMPCMNTRC) command. «

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 trace data for network server description traces can only be transferred to a spooled file. The trace data cannot be transferred to an output file. There are no formatting options available.

Required Parameters

CFGOBJ

Specifies the name of the configuration object being traced. The object is either a line description, a network interface description, or a network server description. » Either the CFGOBJ and CFGTYPE parameters or the FROMSTMF parameter must be specified. «

CFGTYPE

Specifies the type of configuration description that was traced. » Either the CFGOBJ and CFGTYPE parameters or the FROMSTMF parameter must be specified. «

***LIN**: The type of configuration object is a line description.

***NWI**: The type of configuration object is a network

***NWS**: The type of configuration object is a network server description.

» FROMSTMF

Specifies the path name of the stream file from which communications trace data is formatted. This file must have been created by running the Dump Communications Trace (DMPCMNTRC) CL command. Either the CFGOBJ and CFGTYPE parameters or the FROMSTMF parameter must be specified.

from-stream-file-path-name: Specify the path name of the stream file previously created by running the DMPCMNTRC command. «

Optional Parameters

OUTPUT

Specifies the format of the output data.

Note:

For network server description traces, *PRINT must be specified for this parameter.

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

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

OUTFILE

Specifies the database file where the output file is stored.

The name of the output 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 output file name.

OUTMBR

Specifies the name of the database file member to which the output is directed.

Element 1: Member to Receive Output

***FIRST:** The first member in the database file is used.

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

Element 2: Operation to Perform on Member

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

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

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).

***CALC:** The system determines whether to format the user data in EBCDIC or ASCII, based on the type of controller that is used.

***ASCII:** The user data is formatted in ASCII.

***EBCDIC:** The user data is formatted in EBCDIC.

SLTLIND

Specifies whether to format data for all lines or a specific line communicating on the network during a trace.

***ALL:** Formats the data for all lines.

line-name: Specify the line name.

SLTCTLD

Specifies whether to format data for all controllers or a specific controller communicating on the network during a trace.

***ALL:** Formats data for all controllers.

controller-name: Specify the controller name.

FMTSNA

Specifies whether line protocol data or Systems Network Architecture (SNA) data is formatted. Line protocol data includes SDLC, X.25, Carrier Sense Multiple Access with Collision Detection (CSMA/CD), Ethernet DIX V2, DDI, wireless, and IBM Token-Ring Network (TRLAN).

***NO:** Only line protocol data is formatted.

***YES:** Only SNA data is formatted.

FMTRR

Specifies whether receiver ready (RR) and receiver not ready (RNR) commands are formatted with other data.

***NO:** RR and RNR commands are not formatted with other data.

***YES:** RR and RNR commands are formatted with other data.

FMTTCP

Specifies whether line protocol data or Transmission Control Protocol/Internet Protocol (TCP/IP) data is formatted.

***LINTYPE:** For X.25, Ethernet, DDI, wireless, Token-Ring, and Frame Relay lines, only line protocol data is formatted. For all other lines supporting TCP/IP, TCP/IP data is formatted.

***YES:** TCP/IP data is formatted.

***NO:** TCP/IP data is not formatted.

FMTUI

Specifies whether line protocol data or unnumbered information (UI) data is formatted.

***NO:** All line protocol data is formatted.

***YES:** Only UI data is formatted.

FMTMAC

Specifies whether line protocol data or medium access control (MAC) or station management (SMT) data is formatted.

***NO:** The line protocol data is formatted.

***YES:** Only MAC or SMT data is formatted.

FMTETH

Specifies whether IEEE 802.3 data or Ethernet V2 data is formatted.

***YES:** IEEE 802.3 data and Ethernet V2 data are formatted.

***NO:** Only IEEE 802.3 data is formatted.

FMTCCD

Specifies whether all network interface data or only Integrated Services Digital Network (ISDN) signalling data is formatted.

***NO:** All network interface data is formatted.

***YES:** Only Integrated Services Digital Network (ISDN) signaling data is formatted.

FMTBCD

Specifies whether broadcast data and data received containing destination MAC addresses is formatted.

***YES:** Broadcast data is formatted.

***NO:** Broadcast data is not formatted.

EXCLMI

Specifies whether to exclude local management interface (LMI) data from the formatted output.

***NO:** LMI data is not excluded from the formatted output.

***YES:** LMI data is excluded from the formatted output.

FMTLMI

Specifies whether LMI data is formatted.

***NO:** LMI data is not formatted.

***YES:** LMI data is formatted.

Note:

You cannot specify *YES on both the EXCLMI and FMTLMI parameters.

TCPIPADR

Specifies an internet address pair for which TCP/IP data is formatted. Any values that are valid for IP address 1 are also valid for IP address 2.

The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

➤ For IPv6 (IP version 6) addresses, the form is *xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx*, where *x* is any valid hexadecimal digit 0 through F.

Note: IPv6 addresses are only valid when formatting from an IFS file. ⏪

Element 1: IP Address 1

***ALL:** The communications between the systems specified on the IP address 2 element and all other systems are printed.

IP-address-1: Specify the address of the system

SLTPORT

Specifies whether data for all internet protocol (IP) ports or only a single IP port is formatted.

Note: This parameter is valid only if FMTTCP(*YES) is specified

***ALL:** Data for all IP ports is formatted.

IP-port: Specify the IP port number (1 to 65535) whose data is formatted.

FMTLCP

Specifies whether Link Control Protocol (LCP) data is included in the formatted communications trace.

Note: If FMTLCP, FMTNCP, and FMTTCP are all specified *NO when formatting data for a Point-to-Point Protocol (PPP) line, then asynchronous and unrecognized data will be placed in the spooled file. This is also the case if all are specified *YES (or *LINTYPE for FMTTCP). In all other cases asynchronous and unrecognized data will be omitted.

***YES:** LCP data is formatted.

***NO:** LCP data is not formatted.

FMTNCP

Specifies whether Network Control Protocol (NCP) data is included in the formatted communications trace.

***YES:** NCP data is formatted.

***NO:** NCP data is not formatted.

Example for PRTCMNTRC

```
PRTCMNTRC CFGOBJ(*QESLINE) CFGTYPE(*LIN)
```

This command prints communications trace data for line description QESLINE.

Error messages for PRTCMNTRC

*ESCAPE Messages

CPF2634

Not authorized to object &1.

CPF39AF

Trace is ending - please wait

CPF39A7

Trace storage not available in communications processor

CPF39A8

Not authorized to communications trace service tool

CPF39A9

Error occurred during communications trace function

CPF39BA

Formatting options selected not valid

CPF39BB

Communications trace data not printed

CPF39BC

Communications trace print request cannot be completed

CPF39B0

No communications traces exist.

CPF39B1

Trace &1 type &2 does not exist

CPF39B3

Trace &1 type &2 contains no data

CPF39B4

Trace data for &1 type &2 cannot be printed

CPF39B5

Communications trace data not printed

CPF39B6

Communications trace function cannot be performed

CPF39B7

Trace data for &1 type &2 cannot be printed

CPF39B8

No SNA data found in trace &1 type &2

CPF39B9

No trace records found for printing trace &1 type &2

CPF39C4

IP address not valid.

» CPF3CF2

Error(s) occurred during running of &1 API. «

CPF9803

Cannot allocate object &2 in library &3.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

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

CPF9860

Error occurred during output file processing.

» CPF9872

Program or service program &1 in library &2 ended. Reason code &3.

CPFA0D4

File system error occurred. «

PRTCPTTRPT (Print Component Report) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

PRTCPTTRPT Command syntax diagram

Purpose

The Print Component Report (PRTCPTTRPT) command produces a report that expands on the detail for each component of system performance shown on the System Report. This report is produced from the performance data collected by Collection Services and shows the data by job, user, pool, disk, IOP, local work station, exception, database journaling, and TCP/IP. This report is written to the printer file QPPTCPTTR. Jobs may be selectively included in the report or excluded from the report based on a variety of job details and interval times.

Required Parameter

MBR Specifies the performance data member that is used. This name should correspond to the member name specified on the TOMBR parameter of the Create Performance Data (CRTPFDRDTA) command.

Optional Parameters

LIB Specifies the library where the performance data is located.

QPFRDATA: The IBM-supplied performance data library, QPFRDATA, is used to locate the database files.

library-name: Specify the name of the library where the database files are located.

TITLE Specifies the title for the report that is created.

***MBR:** The text of the database member, which contains the performance data, is the report title.

'report-title': Specify the title for the report. Specify up to 50 characters, enclosed in apostrophes.

PERIOD

Specifies the period of time on which to report. The value consists of two lists of two elements each:

PERIOD((start-time start-date)
(end-time end-date))

*N may be used to maintain the position in the parameter value sequence in place of an element that precedes the values that are specified. For example, PERIOD(*N (*N 091290)) specifies the ending date and uses the defaults for the other values.

Element 1: Starting Time

One of the following values is used to specify the starting time. Data collected prior to this time on the starting date is not included in the report.

***FIRST:** Data records starting from the beginning of the first day (00:00:00) of the collection period are included.

***SELECT:** Displays an interval selection screen where intervals are selected for inclusion. This value is valid only in the interactive environment. If this value is used, the remaining values of the PERIOD parameter (start-date, end-time, end-date) are ignored.

start-time: Specify the time of the first data record to include in the report. 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.

All time and date entries must be two digits. This means you must include zeros where appropriate.

Element 2: Starting Date

One of the following values is used to specify the starting date. Data collected prior to the starting time on this date is not included in the report.

***FIRST:** Data records starting from the first day of the collection period are included.

start-date: Specify the date of the first data record to include in the report. The date must be entered in the format specified by QDATFMT and, if separators are used, QDATSEP. For instance, the system might have a date format of 'mm/dd/yy'. The month (mm), day (dd), and year (yy) are all required 1- or 2-digit values. The slashes (/) are optional if all six digits are specified. If the slashes are omitted, or if the value is entered from the prompt display, the apostrophes (') are not required.

Element 3: Ending Time One of the following values is used to specify the ending time. Data collected after this time on the ending date is not included in the report.

***LAST:** Data records through the end of the day (23:59:59) are included in the report.

end-time: Specify the time of the last data record to include in the report. See *start-time* in this parameter for details on how the time must be specified.

Element 4: Ending Date One of the following values is used to specify the ending date. Data collected after the ending time on this date is not included in the report.

***LAST:** Data records through the last day of the collection period are included in the report.

end-date: Specify the date of the last data record to include in the report. Use the same date format used for the starting date.

TYPE Specifies the sections of the report that you want to print.

***ALL** All sections of the report are printed.

***INTERVAL** Specifies that you want to print the Component Interval Activity section.

***WORKLOAD** Specifies that you want to print the Job Workload Activity section.

***POOL** Specifies that you want to print the Storage Pool Activity section.

***DISK** Specifies that you want to print the Disk Activity section.

***IOP** Specifies that you want to print the IOP Utilization Activity section.

***LCLWS** Specifies that you want to print the Local Work Stations section.

***RMTWS** Specifies that you want to print the Remote Work Stations section. You can print this section only if you converted the performance data that was collected by the STRPFMON command in a previous release. Collection Services does not collect this data.

***EXCEPTION** Specifies that you want to print the Exception Occurrence Summary and Interval Counts section.

***DBJRN** Specifies that you want to print the Database Journaling Summary section.

***TCPIP** Specifies that you want to print the TCP/IP Activity section. It will not be available if you convert data from a previous release.

***HTTP** Specifies that you want to print the HTTP Server Activity section. This section includes statistics for HTTP Server (powered by Apache).

DETAIL

Specifies whether you want the report to provide detailed job information at the job level or the thread level.

***JOB** Specifies that you want detailed information at the job level.

***THREAD** Specifies that you want detailed information at the thread level.

SLTJOB

Specifies a list of up to 50 job identifiers to select. Only specified jobs are included in the report.

A job identifier is either the special value *ALL or a qualified name with up to three elements, for example:

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

Note: The SLTJOB and OMTJOB parameters are mutually exclusive.

Element 1: Job name

***ALL:** All jobs are included, unless excluded by some other selection criterion.

job-name: Specify the specific or generic name of the job to select. Because jobs may have identical job names, this value may not identify a specific job. You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

user-name: Specify the user name of the job to select. Because jobs may have identical user names, this value may not identify a specific job.

job-number: Specify the 6-digit number of the job to select. All six digits must be specified; use leading zeros if necessary.

Element 2: Thread

***ALL**: All threads are included, unless excluded by some other selection criterion.

thread-identifier: Specify the thread identifier to select. Because some jobs can have identical thread identifiers, this value may not identify a specific thread.

OMTJOB

Specifies a list of up to 50 jobs to omit. All jobs specified are excluded from the report.

Individual jobs are identified by a job identifier. A job identifier is either the special value *NONE or a qualified name with up to three elements: a job number, a user name, and a job name. Job identifiers are written in the form: job-number/user-name/job-name, but you do not have to specify all three elements. *N can be used in place of an element to maintain the position in the parameter value sequence.

Note: The SLTJOB and OMTJOB parameters are mutually exclusive.

Element 1: Job name

***NONE**: Jobs are not excluded based on job identifier.

job-name: Specify the specific or generic name of the job to omit. Because jobs may have identical job names, this value may not identify a specific job. You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

user-name: Specify the user name of the job to omit. Because jobs may have identical user names, this value may not identify a specific job.

job-number: Specify the 6-digit number of the job to omit. All six digits must be specified; use leading zeros if necessary.

Element 2: Thread

***ALL**: All threads are included, unless excluded by some other selection criterion.

thread-identifier: Specify the thread identifier to select. Because some jobs can have identical thread identifiers, this value may not identify a specific thread.

SLTUSRID

Specifies a list of up to 50 user names to select. Only jobs with one of the specified user names are included in the report.

***ALL:** Jobs with all user names are included, unless excluded by some other selection criterion.

user-name: Specify the user names of the jobs to select. Because jobs may have identical user names, this value may not identify a specific job. SLTUSRID(user) is equivalent to SLTJOB(*N/user/*N). You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

OMTUSRID

Specifies a list of up to 50 user names to omit. Jobs with any of the user names specified are excluded from the report.

***NONE:** Jobs are not excluded based on user name.

user-name: Specify the user names of the jobs to omit. Because jobs may have identical user names, this value may not identify a specific job. OMTUSRID(user) is equivalent to OMTJOB(*N/user/*N). You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

SLTJOBTYPE

Specifies a list of up to 15 job types to be included in the report.

***ALL:** Jobs of all types are included, unless excluded by another selection value.

job-type: Specify the type of job to select. The possible values include:

- A: automatic start jobs
- B: batch jobs
- C: iSeries Access jobs
- D: DDM serverjobs
- E: evoke jobs
- I: interactive jobs
- L: Licensed Internal Code jobs
- M: subsystem monitor jobs
- P: pass-through jobs
- R: spool reader jobs
- S: system jobs
- T: multiple requester terminal (MRT) jobs
- W: spool writer jobs
- X: start system job
- 3: System/36 jobs

OMTJOBTYPE

Specifies a list of up to 15 job types to be omitted from the report.

***NONE:** No job types are excluded, unless excluded by another selection value.

job-type: Specify the type of job to omit. The possible values include:

- A: automatic start jobs
- B: batch jobs
- C: iSeries Access jobs
- D: DDM serverjobs
- E: evoke jobs
- I: interactive jobs
- L: Licensed Internal Code jobs
- M: subsystem monitor jobs
- P: pass-through jobs
- R: spool reader jobs
- S: system jobs
- T: multiple requester terminal (MRT) jobs
- W: spool writer jobs
- X: start system job
- 3: System/36 jobs

SLTJOBPTY

Specifies a range of run priorities to select. Only jobs that ran with a priority in the specified range will be included in the report.

***ALL:** All jobs are included, unless excluded by another selection value.

Element 1: Highest job run priority

highest-priority: Specifies the highest run priority to select. Priorities can be 0 through 99, where 0 is the highest job priority and 99 is the lowest job priority.

Element 2: Lowest job run priority

lowest-priority: Specifies the lowest run priority to select. Priorities can be 0 through 99, where 0 is the highest job priority and 99 is the lowest job priority.

OMTJOBPTY

Specifies a range of run priorities to omit. Only jobs that ran with a priority in the specified range will be excluded from the report.

***NONE:** No jobs are excluded based on their priority.

Element 1: Highest job run priority

highest-priority: Specifies the highest run priority to omit. Priorities can be 0 through 99, where 0 is the highest job priority and 99 is the lowest job priority.

Element 2: Lowest job run priority

lowest-priority: Specifies the lowest run priority to omit. Priorities can be 0 through 99, where 0 is the highest job priority and 99 is the lowest job priority.

SLTPOOLS

Specifies a list of up to 64 pools to select. Only jobs that run in one of the specified pools are included in the report.

***ALL:** Jobs in all pools are included, unless excluded by other selection criteria.

storage-pool-identifier: Specify the number of a pool to select. Valid values range from 1 through 64.

OMTPOOLS

Specifies a list of up to 64 pools to omit. Jobs that run in any of the specified pools are excluded from the report.

Note: The SLTPOOLS and OMTPOOLS parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on their pool.

storage-pool-identifier: Specify the number of the pool to omit. Valid values range from 1 through 64.

SLTSBS

Specifies a list of up to 50 subsystems to select. Only jobs that ran in one of the specified subsystems are included in the report.

Note: The SLTSBS and OMTSBS parameters are mutually exclusive.

***ALL:** Jobs in all subsystems are included unless excluded by other selection criteria.

subsystem-name: Specify the name of a subsystem to select.

OMTSBS

Specifies a list of up to 50 subsystems to omit. Jobs that run in any of the specified subsystems are excluded from the report.

Note: The SLTSBS and OMTSBS parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on subsystem.

subsystem-name: Specify the name of a subsystem to omit.

SLTLINE

Specifies a list of up to 50 communications lines to select. Only jobs that use a remote device connected through one of the specified communications lines are included in the report.

Note: The SLTLINE and OMTLINE parameters are mutually exclusive.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

communications-line-name: Specify the name of a communications line to select. This excludes jobs using remote devices connected through other communications lines (or no communications line), even if the controllers to which these devices are attached are specified on the SLTCTL parameter.

OMTLINE

Specifies a list of up to 50 communications lines to omit. Jobs that use a remote device connected through any of the specified communications lines are excluded from the report.

Note: The SLTLINE and OMTLINE parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on communications line.

communications-line-name: Specify the name of a communications line to omit.

SLTCTL

Specifies a list of up to 50 communications controllers to select. Only jobs that use a device connected to one of the specified communications controllers are included in the report.

Note: The SLTCTL and OMTCTL parameters are mutually exclusive.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

controller-name: Specify the name of a communications controller to select.

OMTCTL

Specifies a list of up to 50 communications controllers to omit. Jobs that use a device connected to any of the specified communications controllers are excluded from the report.


Note: The SLTCTL and OMTCTL parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on communications controllers.

controller-name: Specify the name of a communications controller to omit.

SLTFCNARA

Specifies a list of functional areas to select. Only jobs and users identified in one of the functional areas are included in the report.


A functional area is a list of job names and user names previously defined by the user. Refer to the Performance Tools for iSeries  book for more information on functional areas.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

functional-area-name: Specify the name of the functional area to select.

OMTFCNARA

Specifies a list of functional areas to omit. Jobs and users identified in any of the functional areas are excluded from the report.

A functional area is a list of job names and user names previously defined by the user. Refer to the Performance Tools for iSeries  for more information on functional areas.

***NONE:** Jobs are not excluded based on functional area.

functional-area-name: Specify the name of the functional area to omit.

JOB Specifies the job name used if the job is submitted for batch processing.

Note: If *NONE is specified on the JOBD parameter, this parameter is ignored; job processing is performed interactively.

PRTCPTRPT: The command name is used for the job name.

***MBR:** The name selected for the performance data member on the MBR parameter is used.

job-name: Specify the name used for batch jobs.

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 to be searched.

QPFRJOBD: The IBM-supplied Performance Tools job description QPFRJOB is used.

job-description-name: Specify the name of an alternate job description.

Other Single Values

***NONE**: A batch job is not submitted; processing continues interactively while the user waits. The user's work station is not available for other use during this time, which could be significant for long jobs.

Examples for PRTCPT

Example 1: Printing a Component Report

```
PRTCPT MBR(APRIL18)
```

This command prints a complete component report for the performance data member APRIL18 in library QPFRDATA. The report title is the same as the text in the member.

Example 2: Printing a Report With a Title

```
PRTCPT MBR(NOV1) PERIOD(*SELECT)
  TITLE('Intervals with Highest Response Times')
```

This command prints a component report for the data member NOV1 in library QPFRDATA. The user is presented with the interval-selection display, which allows sorting of the intervals according to various criteria and selection of only certain intervals to be included in the report. The title of the report is *"Intervals with Highest Response Times"*.

Error messages for PRTCPT

*ESCAPE Messages

PCY0013

Performance data files are not upward compatible.

PCY0014

Performance data files are not downward compatible.

PFR1010

Cannot process request because of missing data.

PFR3002

Cannot print report because of missing data.

PFR3004

Incorrect measurement interval specified.

PFR3006

Measurement interval specified is not valid.

PFR3111

Functional area &1 does not exist.

PFR5501

Performance data files are not upward compatible.

PFR5502

Performance data files are not downward compatible.

PFR9048

Cannot display graph because of missing data.

PRTDEVADR (Print Device Addresses) Command Description

PRTDEVADR Command syntax diagram

Purpose

The Print Device Addresses (PRTDEVADR) command provides a printed list of addresses and related information for devices attached to a local or remote work station controller. For each device attached to the work station controller named in the controller description (CTLD parameter), the output shows the device's name, its port and switch setting, its type and model number, shared session number (valid only if device type is 3486 or 3487), and whether the device is a display station or printer.

Required Parameter

CTLD

Specifies the name of the local or remote work station controller for which device address information is printed.

Example for PRTDEVADR

```
PRTDEVADR CTLD(CTL01)
```

This command prints device address information for the devices that are attached to the CTL01 work station controller.

Error messages for PRTDEVADR

*ESCAPE Messages

CPF2602

Controller &1 not found.

CPF2625

Not able to allocate object &1.

CPF2628

Device description previously deleted.

CPF263B

Controller &1 not a work station controller.

CPF2634

Not authorized to object &1.

CPF2778

Controller description &1 damaged.

CPF9846

Error while processing file &1 in library &2.

CPF9850

Override of printer file &1 not allowed.

PRTDSKINF (Print Disk Information) Command Description

PRTDSKINF Command syntax diagram

Purpose

The Print Disk Information (PRTDSKINF) command is used to print disk space information that is already stored in the database file QAEZDISK » or QAEZDnnnnn by the Retrieve Disk Information (RTVDSKINF) command, where 'nnnnn' is the ASP number of the independent ASP for which disk space information was retrieved. « The output with file name QPEZDISK goes to the spool queue associated with the job using this command.

Required Parameter

RPTTYPE

Specifies the type of report to print. The report information is taken from member QCURRENT in QAEZDISK » or QAEZDnnnnn, where 'nnnnn' is the ASP number of the independent ASP for which disk space information was retrieved. « If QCURRENT does not contain any data, an error message is sent.

» 1. If option *FLR is specified on the RPTTYPE parameter, *SYSBAS must be specified in the ASPDEV parameter. Folders are not allowed on auxiliary storage pool (ASP) devices, they are only allowed on the system ASP and basic ASPs. «

*LIB: A report of the library information contained in the file is printed.

*FLR: A report of the folder information contained in the file is printed.

*OWN: A report of the user profile (owner) information contained in the file is printed.

*OBJ: A report of object information contained in the file is printed.

*SYS: A report of only the system information contained in the file is printed.

Optional Parameters

» ASPDEV

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

The possible values are:

*SYSBAS: Disk information for the system ASP and all basic ASPs is printed. File QAEZDISK in library QUSRSYS contains the disk space information that is to be printed.

auxiliary-storage-pool-device-name: Specify the name of the ASP device for which disk space information is to be printed. File QAEZDnnnnn in library QUSRSYS contains the disk space information that is to be printed, where 'nnnnn' is the ASP number of the specified ASP device. «

LIB Specifies the names of the libraries to print information about.

*ALL: The report has information on all user libraries on the system.

library-name: Specify the user library.

generic-library-name*: Specify the generic library 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.

OWNER

Specifies the names of the owners (user profiles) to print information about.

*ALL: The report contains information on all user profiles on the system.

owner-name: Specify the user profile.

generic-owner-name*: Specify the generic user profile.

FLR Specifies the names of the folders to print information about.

***ALL:** The report has information on all user folders on the system.

folder-name: Specify the folder name.

generic-folder-name:* Specify the generic folder name.

DOC Specifies the names of the documents to print information about.

***ALL:** The report contains information on all documents in the specified folder.

document-name: Specify the document by the given name within the specified folder.

generic-document-name:* Specify the documents specified by the generic qualification.

OBJ Specifies the names of the objects to print information about.

***ALL:** If you specify a library or owner, then the object information is all objects within the library or those controlled by the owner

***NONE:** No library or owner is specified.

object-name: Specify a library or owner, then the object information is the object specified by the given name within the library or controlled by the owner.

generic-object-name:* Specify a library or owner, then the object information are the objects that meet the specified generic qualification within the library or controlled by the owner.

OBJTYPE

Specifies the object types to print information about.

***ALL:** If you specify a library or owner, information is printed on all the specified object types within the library or controlled by the owner. If an object name is specified, information on all object types within that name, within the library, or controlled by the owner is printed. If a library or owner is not specified, the report has information on all object types on the system. If an object name is specified, information only on object types with that name is printed.

object-type: Specify a library or owner, then the object type information is the object type specified within the library or controlled by the owner. If an object is specified, the report has information on the objects with the specified object type within the library or controlled by the owner.

MINSIZE

Specifies the size of the smallest piece of information to include. For example, if a library report is requested without objects, then this size would be the size of the smallest library to include. If objects within the library are requested, then this would be the size of the smallest object within the library to include.

0: All objects are included regardless of size.

size: Specify size in thousands of bytes.

SORT Specifies the order in which the information should be sorted.

***SIZE:** Information is sorted from large to small.

***OWNER:** The information is sorted in alphabetical order by owner name.

***LSTCHG:** The information is sorted by last-change date with the oldest information first.

***LSTUSE:** The information is sorted by last-use date with the oldest information first.

***NAME:** Information is sorted in alphabetical order according to the report type.

Example for PRTDSKINF

```
PRTDSKINF ASPDEV(*SYSBAS) RPTTYPE(*LIB) LIB(*ALL) OBJ(*ALL)
SORT(*SIZE)
```

This command prints a library report from database file QAEZDISK in library QUSRSYS in member QCURRENT, containing information about all libraries, objects, and object types in the libraries. The information is sorted by size and sent to the printer file QPEZDISK.

Error messages for PRTDSKINF

*ESCAPE Messages

CPF1ED0

Current collection of disk space information not found.

CPF1ED1

Not authorized to collect disk space information.

CPF1ED2

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

CPF1EEC

Not authorized to file >> &1. <<

CPF1E99

Unexpected error occurred.

PRTDOC (Print Document) Command Description

PRTDOC Command syntax diagram

Purpose

The Print Document (PRTDOC) command permits the user to print a document using the word processing function of OfficeVision.

This command also permits the user to override all print option values that are currently stored with a document. When a document is created, a set of default print options is associated with that document. If the user wants to override one or more of the parameters in this print command, the user must select OPTIONS(*YES) so that the print options appear on the display. When the print options appear, any of the print parameters can be changed. The user can override one or all of the print option parameters with this command. More information on printing documents is in the Using OfficeVision/400 Word Processing book.

Optional Parameters

DOC Specifies the name of the document to print.

*PRV: The name used in the previous session is used.

*ALL: All documents to which the user is authorized are printed to a database file. This is valid only when the output is directed to an OUTFILE.

document-name: Specify the name of the document that is printed.

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

*PRV: The name used in the previous session is used.

folder-name: Specify the name of the folder containing the document to be printed.

OPTIONS

Specifies whether the print options for this document are displayed before the document is printed.

*NO: The print options are not displayed before the document is printed.

***YES:** The print options are displayed before the document is printed. Regularly used print options are set with this special value. For example, if STRPAGE(5) and OPTIONS(*YES) is specified, the value 5 appears on page 1 of the print options display.

***PRTFILE:** The print options specified in the PRTFILE parameter are used. When additional parameters are used, those parameters that are overridden by the appropriate printer file parameters when the document actually prints are not displayed.

Note: When additional parameters are used, those parameters not relevant to outfile processing are not displayed.

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

PRTFILE

Specifies which printer file to use for the print options. The values found in the printer file for the print device, print quality, duplex, output queue, form type, copies, and hold override the corresponding values in the print options for the document. This parameter is valid only if OPTION(*PRTFILE) is also specified.

QSYSVRT: The document is printed using the system printer. This value overrides the printer name specified in the print options associated with the document.

The name of the printer device 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.

printer-device-file-name: Specify the name of the printer file to use for this PRTDOC request. This value overrides the printer file name specified in the print options associated with the document.

OUTFILE

Specifies the qualified name of the database file where the displayed information is stored. If the specified file does not exist, this command creates a database file and file member. If the file is created, the text is labeled 'OUTFILE for PRTDOC' and the public authority is *EXCLUDE. Output to the database file is only supported if the OPTIONS (*OUTFILE) is also specified.

***PRV:** The library and database file used in the previous request is used.

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.

database-file-name: Specify the qualified name of the database file in which the resolved document information is stored.

OUTMBR

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

Element 1: Member to Receive Output

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

***PRV:** The name used in the previous session is used.

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

Element 2: Operation to Perform on Member

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

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

CURSTS

Specifies the value the document Interchange Document Profile (IDP) status field must have before the document may be printed to the database file. This field is 20 characters long and is valid only if OUTFILE output is requested.

Note:

If the name of the document is specified in lowercase letters, the iSeries 400 automatically shifts the name to uppercase letters. If the document name is to remain in lowercase letters, the name must be enclosed in apostrophes.

***PRV:** The name used in the previous session is used.

***NOCHK:** The status field is not checked before printing this document to a database file.

value: Specify the value to which the status field must be equal before the document is printed to the database.

NEWSTS

Specifies the value to which the document IDP status field value is set after the document has been printed to the database file. If a NEWSTS value is specified, the user must have at least *CHANGE authorization to the document. This field is 20 characters long and is valid only if OUTFILE output is requested.

***PRV:** The name used in the previous session is used.

***NOCHG:** The status field is not changed after printing this document to a database file.

value: Specify the value to which the status field is set after the document is printed to a database file.

OUTDTATYP

Specifies whether the entire document, or just the IDP information, is printed to the database file. This is valid only if OUTFILE output is requested.

***PRV:** The name used in the previous session is used.

***ALL:** The entire document is printed to a database file.

***IDP:** Only the IDP information is printed to a database file.

DLTDOC

Specifies whether a document is deleted after it has been printed to the database file. This is valid only if OUTFILE output is selected.

Note:

The user must be the owner of the document or have *ALL authority to delete it.

***NO:** The document is not deleted after being printed to the database file.

***YES:** The document is deleted after being printed to the database file.

OUTPUT

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

***SAME:** The output device specified in the document print options does not change.

*****: The document is shown on the display.

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

DEV Specifies the name of the selected printer.

***SAME:** The name of the printer specified in the document print options does not change.

***USRPRF:** The printer ID indicated in the user profile is used to print the document.

***SYSVAL:** The value specified in the system value QPRTDEV is used.

***WRKSTN:** The printer assigned to the user's work station is used to print the document.

printer-name: Specify the name of the printer to use to print the document.

OUTQ Specifies the qualified name of the output queue.

***SAME:** The output queue value specified in the document print options does not change.

***DEV:** The output queue specified on the PRTDEV parameter is used.

***FILE:** The output queue and output queue library values are based on:

1. If the PRTFILE parameter is specified, the values from the specified printer device are used.
2. If the PRTFILE parameter is not specified, the values from the printer file prompt on the document print options are used.

***WRKSTN:** The output queue assigned to the user's work station is used.

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 to use to hold the output until it is ready to print.

SPLFILE

Specifies the name of the output file in which spooled files are kept.

***SAME**: The name of the output file specified in the document print options does not change.

***FILE**: The name chosen for the output file is the name for the printer file being used.

***DOC**: The document name is used for the spool file name. However, if the document name is longer than 10 characters or contains a period, the spool file name is QSYSPRT.

spool-file-name: Specify the name of the file to which to send the output. The file is then spooled to the queue.

FORMTYPE

Specifies the type of form on which the output is printed. The identifiers used to indicate the type of forms are user-defined and can be a maximum of 10 characters in length.

***SAME**: The type of form specified in the document print options does not change.

***STD**: The standard form type is used. The output is printed on the form type specified in the printer file for the printer selected. The printer file contains the information controlling how the document is printed on a particular printer.

form-type: Specify the type of form to use. Valid values range from 1 through 10 alphanumeric characters.

Note:

Lowercase, blanks, or special characters must be enclosed in apostrophes. For example, a host system form type is entered as FORMTYPE(' '). The value is returned by the host system in a forms mount message.

COVERPAGE

Specifies whether a cover page is printed. The cover page includes such reference items as:

- Document name
- Folder name
- Document description
- Subject
- Reference, and
- Authors' names

***SAME**: The cover page value specified in the document print options does not change.

***YES**: The cover page is printed.

***NO**: The cover page is not printed.

PRTQLTY

Specifies the print quality used to print the document.

***SAME**: The print quality value specified in the document print options does not change.

***LETTER**: The letter quality (highest quality) is used.

***TEXT**: The text quality setting is used.

***DRAFT**: The draft quality (lowest quality) setting is used.

COPIES

Specifies the number of copies to print. This parameter only applies to spooled files.

***SAME:** The number of copies specified in the document print options does not change.

value: Specify a value ranging from 1 through 99 indicating the number of copies to print.

DUPLEX

Specifies whether output is printed on one side or two sides of the paper.

***SAME:** The duplex value specified in the document print options does not change.

***TUMBLE:** The document is printed on both sides of the paper. In addition, this special value indicates that the tops of both sides are on opposite ends of the page.

***YES:** The document is printed on both sides of the paper. In addition, this special value indicates that the tops of both sides are on the same end of the page.

***NO:** The document is printed on one side of the paper.

AUTOBIND

Specifies whether the left and right margins of the even pages will line up with the left and right margins of the odd pages when both sides of the paper are being printed.

***SAME:** The automatic binding value specified in the document print options does not change.

***YES:** The document is adjusted for binding.

***NO:** The document is not adjusted for binding.

HOLD Specifies whether a print job is put on hold. Documents are held on the output queue and can be released to print or deleted. This parameter allows printing of several documents together by putting them on the output queue before releasing them to print.

***SAME:** The hold value specified in the document print options does not change.

***YES:** The print job is held.

***NO:** The print job is not held.

PRTERLOG

Specifies whether a document error log is included as part of the information printed with the document.

***PRV:** The value used in the previous (last) PRTDOC request for this user is used.

***YES:** The error log is printed with the document.

***NO:** The error log is not printed with the document.

ERRFORM

Specifies the type of form on which to print the error log.

***SAME:** The error form value specified in the document print options does not change.

***STD:** The standard form type is used. The output is printed on the form type specified in the printer file for the selected printer. The printer file contains the information controlling how the document is printed on a particular printer.

error-form-name: Specify the name of the form on which to print the error log.

LARGEPRINT

Specifies whether the document is printed using large print.

***SAME:** The large print value specified in the document print options does not change.

***YES:** The document is printed in large print.

***NO:** The document is not printed in large print.

MRGTYPE

Specifies where data is located when it is merged.

***SAME:** The merge source specified in the document print options does not change.

***QRY:** The data is merged from a query. This query is a request to select and copy data from a file of one or more records based on the defined conditions.

***DOC:** The data stored in a document is merged.

***FILE:** The data stored in a file is merged.

***BLANK:** No data is merged.

QRYDFN

Specifies the name of the query that defines the data to be merged. This parameter is valid only when MRGTYPE(*QRY) is selected.

***SAME:** The query name specified in the document print options does not change.

The name of the query 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.

query-definition-name: Specify the query name used to move the merged data.

DTADOC

Specifies the name of the document that contains the data being merged. This parameter is valid only when MRGTYPE(*DOC) is selected.

***SAME:** The document name specified in the document print options does not change.

document-name: Specify the name of the document by selecting a value ranging from 1 to 12 alphanumeric characters. If more than 8 characters are used, the ninth character must be a period (.) followed by a 1 to 3 character extension.

DTAFLR

Specifies the name of the folder that contains the document to merge. This parameter is valid only if MRGTYPE(*DOC) is selected.

***SAME:** The folder name specified in the document print options does not change.

folder-name: Specify the name of the folder that contains the document to merge.

DTAFILE

Specifies the file that contains the member to merge. This parameter is valid only when MRGTYPE(*FILE) is selected.

***SAME:** The file name specified in the document print options does not change.

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

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

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

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

file-name: Specify the name of the file that contains the data to merge.

DTAMBR

Specifies the name of the member that contains the data to merge. This parameter is valid only when MRGTYPE(*FILE) is selected.

***SAME:** The member name specified in the document print options does not change.

***FILE:** The member with the same name as the member name is used.

***FIRST:** The first member is used.

***LAST:** The last member is used.

data-member-name: Specify the name of the member that contains the data to merge.

MLTLINRPT

Specifies whether a multiple line report is created. In this report, data field instructions are merged to create records with several lines of output.

***SAME:** The multiple-line-report option specified in the document print options is used.

***YES:** A multiple line report is created.

***NO:** A multiple line report is not created.

ADJLINES

Specifies whether the line endings in a printed document are adjusted. The line endings are adjusted according to the specifications on the Line Spacing/Justification options display. This parameter is useful when printing a document that has been merged, has instructions, has display attributes that do not print spaces, or that uses a proportionally spaced font.

***SAME:** The line ending adjustment specified in the document print options does not change.

***YES:** The line endings are adjusted.

***NO:** The line endings are not adjusted. This special value is used when text is to be printed out in the same way that it was typed.

ADJPAGES

Specifies whether the page endings in a printed document are adjusted. The page ending adjustment is specified on the first typing line and last typing line prompts on the Page Layout/Paper Options display.

***SAME:** The page ending adjustment specified in the document print options does not change.

***YES:** The page endings are adjusted.

***NO:** The page endings are not adjusted.

ALWWIDOW

Specifies whether the document's page endings are determined by the exact number of lines per page specified on the Page Layout/Paper Options display.

***SAME:** The allow widow lines value specified in the document print options does not change.

***YES:** The page endings are determined by the exact number of lines per page.

***NO:** The page endings are not determined by the exact number of lines per page.

RENUMBER

Specifies whether the pages are renumbered when the document is printed.

***SAME:** The renumber system page numbers value specified in the document print options does not change.

***YES:** The page numbers are renumbered when the document is printed.

***NO:** The page numbers are not renumbered when the document is printed.

PRTCHGSYM

Specifies whether change symbols are printed in the left margin of the document. Change symbols are used to indicate all lines have been revised.

***SAME:** The print-change-symbol value specified in the document print options does not change.

***YES:** The change symbols are printed in the left margin of the document.

***NO:** The change symbols are not printed in the left margin of the document.

SYMBOLS

Specifies whether 5 different kinds of change symbols appear in the left margin of the document.

***SAME:** The change symbol value specified in the document print options does not change.

value: Specify up to 5 change symbol characters to appear in the left margin of the document.

DRAFTSPACE

Specifies whether the spacing value in the document can be adjusted. For example, if 3 (triple) is entered on the Line Spacing prompt, the double spacing value is 6, and 5 blank lines are printed between each line in the text of the document. Nevertheless, the document is still paginated using the value specified in the Line Spacing prompt. Therefore, depending on the amount of text being printed on a page, one page may print over onto a second page.

***SAME:** The draft space value specified in the document print options does not change.

***YES:** The space value in the document is doubled.

***NO:** The space value in the document is not doubled.

LINNBR

Specifies whether line numbers are printed in the document. The line numbers begin with 1 on the first page of the document. Line numbers are not printed in headers or footers.

***SAME:** The line numbers value specified in the document print options does not change.

***YES:** The line numbers are printed in the document.

***NO:** The line numbers are not printed in the document.

RESOLVE

Specifies whether the instructions placed in the document are processed. For example, the Date (.date) instruction is resolved to the actual date, which, in this example, is 04/03/62.

***SAME:** The resolve value specified in the document print options does not change.

***YES:** The instructions placed in the document are resolved.

***NO:** The instructions placed in the document are not resolved. For example, the Date instruction (.date) prints as ***date**.

LEFTSPACES



Specifies whether the left margin of the document is increased.

***SAME:** The left-spaces value specified in the document print options does not change.

value: Specify a value, ranging from 0 through 99, for the number of spaces to add to the left margin in the printed document.

CHRID

Specifies the character identifier (graphic character set and code page) for the file. This parameter allows printing of text that is in different character identifier (graphic character set and code page) coding. The value specified on this parameter is used to instruct the printer device to interpret the hexadecimal byte string to print the same characters that were intended when the text was

created. More information about the character identifier is in the Printer Device Programming  book. A list of valid CHRID values and applicable printers is in the Printer Device Programming  book.

***SAME:** The graphic character set id specified in the document print options does not change.

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

Element 1: Character Set

character-set: Specify the type of graphic character set to use by entering the appropriate 3-digit identifier.

Element 2: Code Page

code-page: Specify the code page value used to create the command parameters. Valid values range from 1 through 999.

SAVOUTPUT

Specifies whether the document being printed is also saved as a final form document.

***SAME:** The save resolved output value specified in the document print options does not change.

***YES:** The printed document is saved as a final form document.

***NO:** The printed document is not saved as a final form document.

SAVDOC

Specifies the name of the document that contains the final form document.

***SAME:** The save document name specified in the document print options does not change.

***BLANK:** A resolved output document is not specified.

document-name: Specify the name of the document that contains the resolved document. The document name must range from 1 to 12 characters in length. If more than 8 characters are selected, the ninth character must be a period (.) followed by a 1 to 3 character extension. If the document name specified does not exist, a document is created.

SAVFLR

Specifies the name of the folder that contains the document being saved in final form.

***SAME:** The save folder value specified in the document print options does not change.

***BLANK:** A resolved output folder is not specified.

folder-name: Specify the name of the folder to contain the final-form document.

JOBQ Specifies whether the print request is put on the job queue.

***SAME:** The place on the job queue that is specified in the document print options does not change.

***YES:** The printing of the document is placed in the job queue.

***NO:** The printing of the document is not placed in the job queue.

JOBID Specifies the name of the job description that describes how the printing job is run.

***SAME:** The place on the job queue that is specified in the document print options does not change.

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.

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

SNDMMSG

Specifies whether a print job has been sent to the job queue and the user wants to receive a message specifying that the job has completed.

***SAME:** The message value specified in the document print options does not change.

***YES:** A message is sent to the user when the print job has completed.

***NO:** A message is not sent to the user when the print job has completed.

CNLERR

Specifies whether printing is stopped if an error is detected within the document.

***SAME:** The cancel error value specified in the document print options does not change.

***YES:** Printing is stopped on the document if an error is detected. An error message stating that the job is canceled is listed in the error log.

***NO:** Printing does not stop if an error is detected.

STRPAGE

Specifies the page on which printing begins.

Note:

If the STRPAGE(*page-number*) value specified is larger than the ENDPAGE(*page-number*) value specified, the entire document is printed.

***PAGERANGE:** The pages specified on the PAGERANGE parameter are printed.

Note:

If the STRPAGE(*page-number*) value specified is larger than the ENDPAGE(*page-number*) value specified, the entire document is printed.

***SAME:** The start page specified in the document print options does not change.

Note:

If STRPAGE(*SAME) is specified and additional page ranges exist in the document print options, an error message is sent and the document is not printed.

***FIRST:** Printing is started on the first page of the document.

***LAST:** Printing is started on the last page of the document.

page-number: Specify the page on which to begin printing. Valid values range from 0.01 through 9999.99.

ENDPAGE

Specifies the page on which printing ends.

***PAGERANGE**: The pages specified on the PAGERANGE parameter are printed.

***SAME**: The end page value specified in the document print options does not change.

Note:

If ENDPAGE(*SAME) is specified and additional page ranges exist in the document print options, an error message is sent and the document is not printed.

***FIRST**: Printing is ended after the first page of the document.

***LAST**: Printing is ended after the last page of the document.

***STRPAGE**: The end page value is the same as the start page value. Only one page is printed.

page-number: Specify the page on which to stop printing. Valid values range from 0.01 through 9999.99.

PAGERANGE

Specifies the page ranges to print. A maximum of 7 ranges can be specified.

***SAME**: The page range specified on the document print options is printed.

Element 1: Start Page

***FIRST**: Printing is started on the first page of the document.

***LAST**: Printing is started on the last page of the document.

page-number: Specify the page on which to begin printing. Valid values range from 0.01 through 9999.99.

Element 2: End Page

***FIRST**: Printing is ended after the first page of the document.

***LAST**: Printing is ended after the last page of the document.

***STRPAGE**: The end page value is the same as the start page value. Only one page is printed.

page-number: Specify the page on which to stop printing. Valid values range from 0.01 through 9999.99.

LBLACROSS

Specifies the number of labels to print across the page.

***SAME**: The label-across-the-page value specified in the document print options does not change.

value: Specify a value, ranging from 1 through 99, that indicates the number of labels to print across the page.

LBLWIDTH

Specifies how wide to make the label. The width of a label is the number of characters from the left edge of the first label to the left edge of the next label, including the blank spaces between the labels. If the width specified is larger than the margins specified for the document, the margins are used as the width.

***SAME**: The label width value specified in the document print options does not change.

value: Specify a value, ranging from 2 through 198, that indicates the label width.

SHEETFEED

Specifies whether sheet feed paper is used for printing and whether there are more than one row of labels on a page. When using sheet feed paper, this is the only parameter to use to print more than one row of labels on a page.

***SAME:** The sheet feed value specified in the document print options does not change.

***YES:** Sheet feed printing is used and there are more than one row of labels on a page.

***NO:** Sheet feed printing is not used, or there is only one row of labels on a page.

LBLDOWN

Specifies the number of rows of labels to print on a page.

***SAME:** The label down value specified in the document print options does not change.

value: Specify a value, ranging from 1 through 99, that indicates the number of rows of labels to be printed on a page.

SHFLEFTMAR

Specifies whether to shift the left margin to prevent text from being truncated.

***SAME:** The SHFLEFTMAR value does not change.

***YES:** When the right margin exceeds the paper edge, the left margin is shifted so that as much text as possible is printed. If the right margin does not exceed the paper edge, the text is not shifted.

***NO:** The left margin is not shifted when text exceeds the right margin. Any text exceeding the right margin is truncated.

Examples for PRTDOC

Example 1: Printing to a File

```
PRTDOC DOC(MYDOC) FLR(MYFLR) OPTIONS(*OUTFILE)
  OUTFILE(MYFILE/MYLIB) OUTMBR(MYMBR *REPLACE)
  CURSTS(*PRV) NEWSTS(*PRV) OUTDTATYP(*PRV)
  PRTERLOG(*PRV) DLTD(*NO)
```

This command prints the document MYDOC in folder MYFLR to the database file MYFILE in library MYLIB in the database file member MYMBR. If the member already exists, it is replaced by the contents of MYDOC. The CURSTS, NEWSTS, OUTDTATYP, and PRTERLOG are taken from the last PRTDOC request. The document is not deleted after it is printed to the database file MYFILE.

Example 2: Printing a Document

```
PRTDOC DOC(MYDOC) FLR(MYFLR) OPTIONS(*NO)
  DEV(MYPRNTR) OUTQ(*DEV)
```

This command prints the document MYDOC in the folder MYFLR on a printer called MYPRNTR.

Example 3: Printing Document Error Log

```
PRTDOC DOC(MYDOC) FLR(MYFLR) OPTIONS(*NO)
  PRTERLOG(*YES)
```

This command prints the document with a document error log attached to it.

Example 4: Increasing Margin

```
PRTDOC DOC(MYDOC) FLR(MYFLR) OPTIONS(*NO)
  LEFTSPACES(10)
```

This command prints the document and has 10 extra spaces inserted in the left margin.

Example 5: Printing a Cover Page

```
PRTDOC DOC(MYDOC) FLR(MYFLR) OPTIONS(*NO)
COVERPAGE(*YES)
```

This command prints the document with a cover page.

Example 6: Printing One Page to a File

```
PRTDOC DOC(MYDOC) FLR(MYFLR) OPTIONS(*OUTFILE)
OUTFILE(MYLIB/MYFILE)
OUTMBR(*FIRST) PAGERANGE((5 5))
```

This command prints page 5 of the document to the database file MYFILE in library MYLIB in the first member.

Error messages for PRTDOC

*ESCAPE Messages

CPF6C01

Error occurred during data stream transformation.

CPF6C03

Error occurred during document conversion.

CPF9012

Start of document interchange session not successful for &1.

CPF9801

Object &2 in library &3 not found.

CPF9810

Library &1 not found.

CPF9820

Not authorized to use library &1.

OFCFFFC

User storage capacity exceeded.

OFCFFFD

Damaged object found.

OFC8EA3

OfficeVision for AS/400 editor is not available to resolve to a display.

OFC8E01

Printer ID selected is not correct.

OFC8E1C

Cannot delay output when spooling is not active.

OFC8E1D

Printer for large print is not correct.

OFC8E2A

Output file member is in use.

OFC8E2B

Not authorized to output disk file or library.

OFC8E2C

Output disk file member could not be opened.

- OFC8E30**
Incorrect character set ID specified.
- OFC8E38**
Member is not a document output file member.
- OFC8E4D**
Output file name &9 is incorrect.
- OFC8E50**
Job has been canceled because of error.
- OFC8E6B**
Not authorized to output disk file member.
- OFC8E6D**
Could not access the output disk file member.
- OFC80B5**
OfficeVision for OS/400 editor is not available on the system.
- OFC800A**
Folder is in use.
- OFC800B**
Document &1 is in use.
- OFC800E**
&1 already exists as document or folder.
- OFC800F**
Display does not support text.
- OFC8006**
Folder not found.
- OFC8007**
Document &1 not found in folder.
- OFC8008**
Request not allowed with folder.
- OFC8009**
Request not allowed with document &1.
- OFC801A**
Document has been saved to diskette, tape or save file.
- OFC8010**
Document &1 cannot be processed.
- OFC8011**
Document &1 needs to be recovered.
- OFC8016**
Document &1 is checked out.
- OFC8018**
Document &1 is empty.
- OFC802C**
Label option specified with non-label document.
- OFC802D**
Option not allowed for PC editor.

- OFC8029**
Cannot save resolved output when printing a resolved document
- OFC820D**
Library &4 was not found.
- OFC820F**
Member &3 is in use.
- OFC947E**
Fill-in document &1 could not be opened.
- OFC9486**
Printer file or printer file library was not found.
- OFC960A**
&1 key was pressed by the user to end the PRTDOC function.
- OFC9609**
&1 is the resolved output file name for the print options function.
- OFC980B**
&9 documents printed, &10 documents not processed.
- OFC980C**
Error printing document &1 to a file.
- OFC980D**
Error converting document &1.
- OFC980E**
Error converting document &1.
- OFC980F**
Could not delete document &1 from folder.
- OFC9801**
Document &1 could not be opened.
- OFC9802**
Folder could not be opened.
- OFC9806**
No documents were selected for printing.
- OFC9808**
Document &1 does not have selected status.
- OFC9809**
Error log incorrect with document descriptions only.
- OFC9810**
Could not update status for document &1.
- OFC9811**
Folder needs to be reclaimed.

PRTERRLOG (Print Error Log) Command Description

PRTERRLOG Command syntax diagram

Purpose

The Print Error Log (PRTERLOG) command is used primarily for problem analysis. It places a formatted printer file of the data in the machine error log in a spooled printer device file named QPCSMPT or in a specified output file. The error log data can be used by the IBM service representatives.

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

Optional Parameters

TYPE Specifies the type of error log data from the machine error log to print in the spooled printer file.

***ALL:** All the error codes in the machine's error log are printed. In addition, the error codes for each subsystem (for example, direct access storage devices or printers) are printed in summary form.

***ALLSUM:** All the data in the error log is printed in summary form.

***ANZLOG:** A one-line summary is created for each entry in the error log.

***MCH:** Only the error data produced by machine checks is printed.

***DEV:** Only the error data produced by the devices specified on the DEV or RSRNAME parameters are printed.

***ERRLOGID:** Only the error data with the specified error log record is printed. If this value is specified, the ERRLOGID parameter must also be specified. It is ignored for other request types.

VOLSTAT:** If this value is specified and if OUTPUT(*PRINT) is specified, the tape or diskette volume 'lifetime' statistical data records are printed. If this value is specified and if OUTPUT(*OUTFILE) is specified, 'session' volume statistical data records are printed. If the name of the volume is reported as '**', it means that this volume is not displayable.

DEV Specifies the device names for which the user wants the error log data printed.

Note: This parameter is valid only if TYPE(*DEV) is specified.
This parameter cannot be specified if the RSRNAME parameter is specified.

***ALL:** The error log data is printed for all device names.

logical-device-name: Specify one or more device names for which error log data is to be printed. Up to ten device names can be specified.

RSRNAME

Specifies the name of the resource for which error log entries are printed. This parameter cannot be specified if the DEV parameter is specified. This parameter is valid only if TYPE(*DEV) is specified. Up to ten resource names can be specified.

Note: If you specify a storage controller input/output processor (IOP) as the resource name, no error log entries are printed for the resource.

ERRLOGID

Specifies that error log entries with the specified error log identifier are printed. This parameter is valid only if TYPE(*ERRLOGID) is specified. It is ignored for other request types. Up to ten error log identifiers can be specified.

PRTFMT

Specifies that the indicated report prints any hexadecimal data in character format. This parameter cannot be specified if TYPE(*VOLSTAT) is specified or if the OUTFILE parameter is specified.

***CHAR:** The report is formatted so that hexadecimal data prints as character data.

***HEX:** The report is formatted so that hexadecimal data is printed in hexadecimal format.

Note:

Specifying PRTFMT(*HEX) can cause spooling or printing of large amounts of data. This can impact overall system performance.

PERIOD

Specifies the period of time for which the error log data is printed. This parameter is specified as two sets of two values each.

Element 1: Start Time

***AVAIL:** The error data that is available for the specified starting date is printed.

start-time: Specify the start time on the specified start date for which the error data is printed. 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: Start Date

***CURRENT:** The error data that is available for the current day and between the specified start and end times (if specified) is printed.

start-date: Specify the starting date. The date must be specified in the job date format.

Element 3: End Time

***AVAIL:** The error data that is available for the specified end date is printed.

end-time: Specify the ending time for the specified ending date that determines the error data that is printed.

Element 4: End Date

***CURRENT:** The error data that is available for the current day and between the specified start and end times (if specified) is printed.

end-date: Specify the end date for which error data is printed. The date must be specified in job date format.

OUTPUT

Specifies whether the output from the command is printed with the job's spooled output or directed to a database file. More information on this parameter is in commonly used parameters.

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

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

OUTFILE

Specifies the database file to which the report is directed. If the output file already exists, the system attempts to use it. Records replace or are added to existing data in the file member. If the output file does not exist, the system creates a database file in the specified library. A member is created for the file with the name specified in the OUTMBR parameter.

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

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

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

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

file-name: Specify the name of the file that will receive the report.

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

VOLTYPE

Specifies the volume type of the specified volume identifier. Valid values are 4-digit device type numbers for cartridge tape, reel tape, or diskette. This parameter returns information about all the volumes that use the same technology as the tape device type that was specified. For example, if 6380 is the specified value for this parameter, information about all 1/4 inch tape cartridges on the system is returned.

MODEL

Specifies the model number of the specified model type. This parameter is required if the device type is 9331 and TYPE(*VOLSTAT) is specified.

VOL

Specifies one or more volume identifiers used by the file. More information on this parameter is in commonly used parameters.

***ALL:** Volume statistics are processed for all volumes.

volume-name: Specify the name of the volume for which statistics are processed. A maximum of 10 volume names can be specified.

VOLSTAT

Specifies whether the volume statistical data records are kept or deleted from the machine error log after they are printed. This parameter is valid only if TYPE(*VOLSTAT) is specified.

Note:

ENDOPT(*UNLOAD) must be specified during the SAVE operation to generate volume statistics at the completion of the tape operation.

***KEEP:** The volume statistical data records are kept in the error log after they are printed.

***DLT:** The volume statistical data records are deleted from the error log for volumes that are not active after they are printed.

Note:

If OUTPUT(*OUTFILE) is specified, *DLT cannot be specified.

VOLSTATTYP

Specifies the type of volume statistics printed or directed to an output file. This parameter is valid only if TYPE(*VOLSTAT) is specified.

***LIFETIME:** Lifetime statistics are printed. Lifetime statistics cannot be placed in an output file.

***SESSION:** Session statistics are directed to the output file specified on the OUTFILE parameter. Session statistics cannot be printed.

SELECT

Specifies which type of error log entries are included on the report.

***ALL:** All error log entries are included on the report.

***PRC:** The processor error log entries are included on the report.

***MEDIA:** The error log entries for disk, tape, and diskette devices are included on the report.

***LWS:** The error log entries for local workstations are included on the report.

***CMN:** The error log entries for communications are included on the report. These include entries for communications I/O processors, I/O adapters, ports, lines, controllers, and devices connected with SDLC, ASYNC, BSC, X.25, IDLC, ISDN, and local area network line protocols.

***PWR:** The error log entries for system power control network (SPCN) are included on the report.

***LPP:** The error log entries for licensed programs are included on the report.

***LIC:** The error log entries for Licensed Internal Code are included on the report.

SORT Specifies the order in which the entries appear on the report.

***DATETIME:** The entries are sorted by date and time. The summary entries are for each day.

***TIME:** The entries are sorted by the time of day only. The summary entries are for each hour.

***DEVADR:** The entries are sorted by the address of the device. The summary entries are divided into three levels: those for which the first two digits of the address are the same, those for which the first four digits of the address are the same, and those for which the first eight digits of the address are the same.

***ERRTYPE:** The entries are sorted by the severity of the type of error. The more severe error types report at the top of the list. The summary entries are divided into two levels: those that have a common error type, and those that have a common error type and system reference code.

***RSRCNAME:** The entries are sorted by the resource name of the device.

Examples for PRTERRLOG

Example 1: Printing Error Log Data

```
PRTERRLOG
```

This command gets the error data in the machine error log that occurred for all device types and puts it in a spooled file. The entire error log is printed and any hexadecimal data is in character format.

Example 2: Using the System Resource Manager Database

```
PRTERRLOG TYPE(*DEV) RSRCTYPE(TAPE000001)
PRTFMT(*HEX)
```

This command uses the system resource manager database to determine the device type, model, and serial number for the resource TAPE000001. The print request is based on that information. The report is put in the spooled file and contains all records that pertain to that device type, model, and serial number. Any hexadecimal data in the file is converted to hexadecimal format.

Example 3: Processing Error Log Entries

```
PRTERRLOG TYPE(*DEV) DEV(DISKLU1)
OUTPUT(*OUTFILE) OUTFILE(MYLIB/MYDBD)
OUTMBR(ELOG)
```

This command processes all the error log entries for the logical device named DISKLU1. They are put in the file MYDBD, in the library MYLIB, and in the member ELOG. No spooled files are created.

Error messages for PRTERRLOG

*ESCAPE Messages

CPD36CA

OUTPUT(*OUTFILE) cannot be specified with DEV(*ALL).

CPD36C2

DEV and RSRCTYPE cannot be used together.

CPD36C3

PRTFMT parameter not valid with TYPE(*VOLSTAT).

CPD36C4

OUTFILE not valid with PRTFMT parameter.

CPD36C5

RSRCTYPE parameter can only be used with TYPE(*DEV) parameter.

CPD36C7

ERRLOGID valid only with TYPE(*ERRLOGID).

CPD36C9

PERIOD not valid for specified TYPE and VOLSTATYP.

CPF3535

Error log not available for printing.

CPF3541

No error log entries were found.

CPF3593

PERIOD parameter start time exceeds end time.

CPF3693

Service function ended because error occurred.

CPI36CA

Resource name &1 not found.

CPI36CC

No error log entries were found for &1 &2.

PRTINTDTA (Print Internal Data) Command Description

PRTINTDTA Command syntax diagram

Purpose

The Print Internal Data (PRTINTDTA) command is used primarily for problem analysis. It writes the machine internal data to a spooled printer file. The data is used to service the system.

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.
2. **»** To use this command you must have *SERVICE special authority, or be authorized to the Service Dump 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_DUMP, can also be used to change the list of users that are allowed to perform dump operations.
3. The command must be issued from within the job with internal data being printed, or the issuer of the command must be running under a user profile which is the same as the job user identity of the job with internal data being printed, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority. The job user identity is the name of the user profile by which a job is known to other jobs. It is described in more detail in the Work management topic in the Information Center. **«**
4. This command is intended for use by service representatives only.

Required Parameter

TYPE Specifies the type of data to be printed.

***DMP:** The data to be printed was dumped by a previously issued Dump Job Internal (DMPJOBINT) command or by the machine when it processes a device error or object damage. The dump identifier of the data must be specified on the DMPID parameter.

***INTCFG:** The machine internal configuration and resource information is printed.

***NOTES:** The notes portion of the machine internal data, for the period specified by the PERIOD parameter, is printed.

» *JOB: The data to print was dumped by the job identified by the JOB parameter. **«**

Optional Parameters

DMPID

Specifies, for internal dump operations only, the dump identifier associated with the machine internal data that is printed. This parameter must be specified *only* if TYPE(*DMP) is specified; otherwise, it is ignored. The dump identifier is sent in one of three ways:

- In a message to the job that issued the DMPJOBINT command that dumped the internal data
- In a damage message
- In a device error message or machine function check message. The message containing the dump identifier may also be sent to the system history log.

***NONE:** No dump identifier is specified.

dump-identifier: Specify the dump identifier of the dump output that is printed. The identifier specified must contain eight characters.

PERIOD

Specifies the period of time for which the notes portion of the machine internal data is printed. This parameter is valid only if TYPE(*NOTES) is specified; otherwise, it is ignored. The following values can be coded in this parameter, which contains two sets of two values each. Refer to the syntax diagram for the location of each value specified. If this parameter is not specified, all the available notes for the current date are printed.

Element 1: Starting Time

***AVAIL:** The notes that are available from the starting date to the ending date (or for the current day only) are printed.

start-time: Specify the starting time for the specified starting date for which the user wants the notes printed. 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

***CURRENT:** The notes that are available for the current day and between the specified starting and ending times (if specified) are printed.

start-date: Specify the date printed. The date must be entered in the format specified by the system values QDATFMT and, if separators are used, QDATSEP.

Element 3: Ending Time

***AVAIL:** The notes that are available from the starting date to the ending date (or for the current day only) are printed.

end-time: Specify the ending time for the specified ending date that determines the notes that is printed.

Element 4: Ending Date

***CURRENT:** The notes that are available for the current day and between the specified starting and ending times (if specified) are printed.

end-date: Specify the ending date after which the notes are no longer printed. The system date format must be used.



JOB Specifies the qualified name of the job for which the data will be dumped. This parameter is valid only if *JOB is specified on the TYPE parameter; otherwise, it is ignored.

*****: The job that issued this command is the job that will be dumped.

job-name: Specify the name of the job that will be dumped.

user-name: Specify the user name that identifies the user profile under which the job was run.

job-number: Specify the system-assigned job number of the job that will be dumped.

SLTTHD

Specifies a list of up to twenty threads in the job whose information is to be included. This parameter is valid only if *JOB is specified on the TYPE parameter; otherwise, it is ignored.

***ALL**: All threads are dumped.

***SELECT**: A list of thread identifiers is shown from which the user can select up to twenty to be included. *SELECT is only valid if the PRTINTDTA command is run in an interactive session; otherwise, an error message is sent.

thread-identifier: Specify the identifiers of up to twenty threads whose information is to be included. A thread identifier is a string of eight hexadecimal characters. For example, SLTTHD(0000010C000003F8) would select two threads to be dumped. <<

Examples for PRTINTDTA

>> Example 1: Dump by Dump Identifier <<

```
PRTINTDTA TYPE(*DMP) DMPID(0102FA3C)
```

This command prints the job internal dump output that has a dump identifier of 0102FA3C.

>> Example 2: Dump by Job Identifier <<

```
PRTINTDTA TYPE(*JOB)
          JOB(201230/ALMATM/QPADEV0008)
          SLTTHD(*ALL)
```

This command prints the job internal dump output for the selected job including all threads information. <<

Error messages for PRTINTDTA

*ESCAPE Messages

>> CPF3517

Cannot specify *SELECT for the thread ID to include.

CPF3519

Cannot start service function.

CPF6801

Command prompting ended when user pressed &1.

<<

PRTIPSCFG (Print IP over SNA Configuration) Command Description

PRTIPSCFG Command syntax diagram

Purpose

The Print IP over SNA Configuration (PRTIPSCFG) command prints information about the current AF_INET Sockets over SNA configuration. The spooled file created by this CL command is named QSYSPRT. It is sent to the job default output queue. The user data value of the spooled file is PRTIPSCFG.

There are no parameters for this command.

Example for PRTIPSCFG

```
PRTIPSCFG
```

This command prints the current AF_INET sockets over SNA configuration data.

Error messages for PRTIPSCFG

*ESCAPE Messages

CPFA116

&1 configuration not printed.

PRTJOBDAUT (Print Job Description Authority) Command Description

PRTJOBDAUT Command syntax diagram

Purpose

The Print Job Description Authority (PRTJOBDAUT) command allows you to print a report of the job descriptions in a library that do not have public authority of *EXCLUDE, and a user name is specified in the job description. This is a way to check for job descriptions that every user on the system is authorized to use that allow the user to run as another user profile.

This command will print two reports for a library. The first report (Full Report) will contain all of the job descriptions that do not have public authority of *EXCLUDE and have a user name specified. The second report (Changed Report) will contain the job descriptions that now do not have public authority of *EXCLUDE or have a user name specified that either did have public authority of *EXCLUDE, did not have a user name specified, or did not exist when the PRTJOBDAUT command was previously run for the library. If the PRTJOBDAUT command was not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional job descriptions do not have public authority of *EXCLUDE and a user name specified, then the 'Changed Report' will be printed but there will be no job descriptions listed. Changes to user profile special authorities will not cause a 'Changed Report' to be generated.

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

Required Parameter

LIB Specifies the name of the library to search for job descriptions with public authority that is not *EXCLUDE and a user name is specified.

***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 auxiliary storage pools (ASPs) associated with the current job are searched. <<

***ALLUSR:** >> All user libraries in the ASPs associated with the current job are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

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

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries and are also searched:

QDSNX	QS36F	QUSROND
QGPL	QUSER38	QUSRPOSGS
QGPL38	QUSRADSM	QUSRPOSSA
QMPGDATA	QUSRBRM	QUSRPYMSVR
QMQMDATA	QUSRDIRCL	QUSRRDARS
QMQMPROC	QUSRDIRDB	QUSRSYS
QPFRDATA	QUSRIJS	QUSRVI
QRCL	QUSRINFSKR	QUSRVxRxMx
QRCLnnnnn	QUSRNOTES	

Notes:

1. "nnnnn" 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.

***ALLAVL:** All libraries in all available ASPs are searched.

***ALLUSRAVL:** All user libraries in all available ASPs are searched. Refer to *ALLUSR for a definition of user libraries. <<

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

Optional Parameter

CHGRPTONLY

Specifies whether just the changed report should be printed.

***NO:** The full and changed reports will be printed.

***YES:** Only the changed report will be printed.

Example for PRTJOBDAUT

PRTJOBDAUT LIB(QGPL)

This command will generate the full and changed report for the job descriptions in the library QGPL.

Error messages for PRTJOBDAUT

*ESCAPE Messages

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

PRTJOB RPT (Print Job Report) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

PRTJOB RPT Command syntax diagram

Purpose

The Print Job Report (PRTJOB RPT) command produces a job-oriented report from the performance data collected by Collection Services. The report, which is written to the printer file QPPTITVJ, shows job information by interval. Jobs are selected for inclusion in, or exclusion from, the report based on a variety of job details and interval times.

Required Parameter

MBR Specifies the performance data member used. This name should correspond to the member name specified on the TOMBR parameter of the Create Performance Data (CRTPF RDTA) command.

Optional Parameters

LIB Specifies the library where the performance data is stored.

QPFRDATA: The performance data is saved in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the library where the performance data is stored.

TITLE Specifies a title that prints in the heading at the top of each page of the report.

***MBRTXT:** The text of the selected member is used.

***BLANK:** No title is printed.

'report-title': Specify a title of up to 50 characters, enclosed in apostrophes.

PERIOD

Specifies the period of time on which to report. The value consists of two lists of two elements each:

```
PERIOD((start-time start-date)
        (end-time end-date))
```

*N may be used to maintain the position in the parameter value sequence in place of an element that precedes the values that are specified. For example, PERIOD(*N (*N 091290)) specifies the ending date and uses the defaults for the other values.

Element 1: Starting Time

One of the following values is used to specify the starting time. Data collected prior to this time on the starting date is not included in the report.

***FIRST:** Data records starting from the beginning of the first day (00:00:00) of the collection period are included.

start-time: Specify the time of the first data record to include in the report. 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.

All time and date entries must be two digits. This means that you must include leading zeros where appropriate.

Element 2: Starting Date

One of the following values is used to specify the starting date. Data collected prior to the starting time on this date is not included in the report.

***FIRST:** Data records starting from the first day of the collection period are included.

start-date: Specify the date of the first data record to include in the report. The date must be entered in the format specified by QDATFMT and, if separators are used, QDATSEP. For instance, the system might have a date format of 'mm/dd/yy'. The month (mm), day (dd), and year (yy) are all required 1- or 2-digit values. The slashes (/) are optional if all six digits are specified. If the slashes are omitted, or if the value is entered from the prompt display, the apostrophes (') are not required.

Element 3: Ending Time One of the following values is used to specify the ending time. Data collected after this time on the ending date is not included in the report.

***LAST:** Data records through the end of the day (23:59:59) are included in the report.

end-time: Specify the time of the last data record to include in the report. See *start-time* in this parameter for details on how the time must be specified.

Element 4: Ending Date One of the following values is used to specify the ending date. Data collected after the ending time in this date is not included in the report.

***LAST:** Data records through last day of the collection period are included in the report.

end-date: Specify the date of the last data record to include in the report. Use the same date format used for the starting date.

Other Single Value

***SELECT:** Displays an interval selection screen where intervals are selected for inclusion. This value is valid only in the interactive environment. If this value is used, the remaining values on the PERIOD parameter (start-date, end-time, end-date) are ignored.

DETAIL

Specifies whether you want the report to provide detailed job information at the job level or the thread level.

***JOB** Specifies that you want detailed information at the job level.

***THREAD** Specifies that you want detailed information at the thread level.

SLTJOB

Specifies a list of up to 50 job identifiers to select. Only specified jobs are included in the report.

Individual jobs are identified by a job identifier. A job identifier is either the special value *ALL or a qualified name with up to three elements: a job number, a user name, and a job name. Job identifiers are written in the form: job-number/user-name/job-name, but you do not have to specify all three elements. *N can be used in place of an element to maintain the position in the parameter value sequence. For example, you can specify:

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

Note: The SLTJOB and OMTJOB parameters are mutually exclusive.

Element 1: Job name

***ALL:** All jobs are included, unless excluded by some other selection criterion.

job-name: Specify the specific or generic name of the job to select. Because jobs may have identical job names, this value may not identify a specific job. You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

user-name: Specify the user name of the job to select. Because jobs may have identical user names, this value may not identify a specific job.

job-number: Specify the 6-digit number of the job to select. All six digits must be specified; use leading zeros if necessary.

Element 2: Thread

***ALL:** All threads are included, unless excluded by some other selection criterion.

thread-identifier: Specify the thread identifier to select. Because some jobs may have identical thread identifiers, this value may not identify a specific job.

OMTJOB

Specifies a list of up to 50 jobs to omit. All jobs specified are excluded from the report.

A job identifier is either the special value *NONE or a qualified name with up to three elements: a job number, a user name, and a job name. Job identifiers are written in the form:

job-number/user-name/job-name, but you do not have to specify all three elements. *N can be used in place of an element to maintain the position in the parameter value sequence.

Note: The SLTJOB and OMTJOB parameters are mutually exclusive.

Element 1: Job name

***NONE:** Jobs are not excluded based on job identifier.

job-name: Specify the specific or generic name of the job to omit. Because jobs may have identical job names, this value may not identify a specific job.

user-name: Specify the user name of the job to omit. Because jobs may have identical user names, this value may not identify a specific job.

job-number: Specify the 6-digit number of the job to omit. All six digits must be specified; use leading zeros if necessary.

Element 2: Thread

***ALL:** All threads are included, unless excluded by some other selection criterion.

thread-identifier: Specify the thread identifier to select. Because some jobs may have identical thread identifiers, this value may not identify a specific job.

SLTUSRID

Specifies a list of up to 50 user names to select. Only jobs with one of the specified user names are included in the report.

***ALL:** Jobs with all user names are included, unless excluded by some other selection criterion.

user-name: Specify the specific or generic user name of the job to select. Because jobs may have identical user names, this value may not identify a specific job. SLTUSRID(user) is equivalent to SLTJOB(*N/user/*N). You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

OMTUSRID

Specifies a list of up to 50 user names to omit. Jobs with any of the user names specified are excluded from the report.

***NONE:** Jobs are not excluded based on user name.

user-name: Specify the specific or generic user name of the job to omit. Because jobs may have identical user names, this value may not identify a specific job. OMTUSRID(user) is equivalent to OMTJOB(*N/user/*N). You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

SLTPOOLS

Specifies a list of up to 64 pools to be included in the report. Only jobs that run in one of the specified pools are included in the report.

***ALL:** Jobs in all pools are included, unless excluded by other selection criteria.

storage-pool-identifier: Specify the number of a pool to select. Valid values range from 1 through 64.

OMTPOOLS

Specifies a list of up to 64 pools to omit. Jobs that run in any of the specified pools are excluded from the report.

Note: The SLTPOOLS and OMTPOOLS parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on the pool

storage-pool-identifier: Specify the number of the pool to omit. Valid values range from 1 through 64.

SLTSBS

Specifies a list of up to 50 subsystems to select. Only jobs that run in one of the specified subsystems are included in the report.

Note: The SLTSBS and OMTSBS parameters are mutually exclusive.

***ALL:** Jobs in all subsystems are included unless excluded by other selection criteria.

subsystem-name: Specify the name of a subsystem to select.

OMTSBS

Specifies a list of up to 50 subsystems to omit. Jobs that run in any of the specified subsystems are excluded from the report.

Note: The SLTSBS and OMTSBS parameters are mutually exclusive.

***NONE:** Jobs are not excluded bases on subsystem.

subsystem-name: Specify the name of a subsystem to omit.

SLTLINE

Specifies a list of up to 50 communications lines to select. Only jobs that use a remote device connected through one of the specified communications lines are included in the report.

Note: The SLTLINE and OMTLINE parameters are mutually exclusive.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

communications-line-name: Specify the name of a communications line to select. This excludes jobs using remote devices connected through other communications lines (or no communications line), even if the controllers to which these devices are attached are specified on the SLTCTL parameter.

OMTLINE

Specifies a list of up to 50 communications lines to omit. Jobs that use a remote device connected through any of the specified communications lines are excluded from the report.

Note: The SLTLINE and OMTLINE parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on communications line.

communications-line-name: Specify the name of a communications line to omit.

SLTCTL

Specifies a list of up to 50 communications controllers to select. Only jobs that use a device connected to one of the specified communications controllers are included in the report.

Note: The SLTCTL and OMTCTL parameters are mutually exclusive.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

controller-name: Specify the name of a communications controller to select.

OMTCTL

Specifies a list of up to 50 communications controllers to omit. Jobs that use a device connected to any of the specified communications controllers are excluded from the report.


Note: The SLTCTL and OMTCTL parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on communications controllers.

controller-name: Specify the name of a communications controller to omit.

SLTFCNARA

Specifies a list of up to 50 functional areas to select. Only jobs and users identified in one of the functional areas are included in the report.


A functional area is a list of job names and user names previously defined by the user. Refer to the Performance Tools for iSeries  book for more information on functional areas.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

functional-area-name: Specify the name of the functional area to select.

OMTFCNARA

Specifies a list of functional areas to omit. Jobs and users identified in any of the functional areas are excluded from the report.

A functional area is a list of job names and user names previously defined by the user. Refer to the Performance Tools for iSeries  book for more information on functional areas.

***NONE:** Jobs are not excluded based on functional area.

functional-area-name: Specify the name of the functional area to omit.

OMTSYSTSK

Specifies whether or not you want to omit printing the system tasks.

***YES:** Print only the user jobs and omit the system tasks.

***NO:** Print the user jobs and the system tasks.

JOB Specifies the job name used for submitting the job if batch processing.

Note: If *NONE is specified on the JOB parameter, this parameter is ignored; job processing is performed interactively.

PRTJOBRPT: The command name is used for the job name.

***MBR:** The name selected for the performance data member in the MBR parameter is used.

job-name: Specify the name to be used for batch jobs.

JOB 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 to be searched.

The possible job description values are:

QPFRJOBD: The IBM-supplied Performance Tools job description, QPFRJOB, is used.

job-description-name: Specify the name of an alternate job description.

Other Single Values

***NONE:** A batch job is not submitted; processing continues interactively while the user waits. The user's work station is not available for other use during this time, which could be significant for long jobs.

Examples for PRTJOB RPT

Example 1: Submitting a Batch Job

```
PRTJOB RPT MBR(DTA071588A)
```

This command submits a batch job to print a report on all jobs in all intervals in the member DTA071588A of the performance data files in library QPFRDATA. The report title is taken from the text of that member.

Example 2: Selecting Intervals to Include in Report

```
PRTJOB RPT MBR(DTA071588A) PERIOD(*SELECT)
```

This command submits a job to print a report from the same data, but first shows a screen where a user interactively selects which intervals to include.

Example 3: Reporting on a Specific Time Period

```
PRTJOB RPT MBR(DTA071588A) PERIOD((2330)(0130))
```

This command submits a job to print a report on the data collected from 11:30 PM on the first day of collection through 1:30 AM on the last day of collection. However, if data collection started and ended on the same day, an error message is printed instead, because the specified ending date and time is before the specified starting date and time.

Example 4: Printing a Report Interactively

```
PRTJOB RPT MBR(DTA071588A) SLTUSRID(D46*)
JOB D(*NONE)
```

This command interactively prints a report for all jobs with a user ID starting with D46.

Example 5: Printing a Report Interactively

```
PRTJOB RPT MBR(DTA071588A) SLTJOB(D46*/*N)
JOB D(*NONE)
```

This command performs the same function as the previous example.

Error messages for PRTJOB RPT

*ESCAPE Messages

PCY0013

Performance data files are not upward compatible.

- PCY0014**
Performance data files are not downward compatible.
- PFR1010**
Cannot process request because of missing data.
- PFR3002**
Cannot print report because of missing data.
- PFR3004**
Incorrect measurement interval specified.
- PFR3006**
Measurement interval specified is not valid.
- PFR3101**
The SLTJOB and OMTJOB parameters are mutually exclusive.
- PFR3102**
The SLTUSRID and OMTUSRID parameters cannot both be specified.
- PFR3103**
The SLTPOOLS and OMTPOOLS parameters cannot both be specified.
- PFR3105**
SLTLINE and OMTLINE parameters cannot both be specified.
- PFR3106**
SLTCTL and OMTCTL parameters cannot both be specified.
- PFR3107**
SLTFCNARA and OMTFCNARA parameters cannot both be specified.
- PFR3108**
SLTLOC and OMTLOC parameters cannot both be specified.
- PFR3111**
Functional area &1 does not exist.
- PFR5501**
Performance data files are not upward compatible.
- PFR5502**
Performance data files are not downward compatible.
- PFR9005**
YAXIS(*TIME) must be specified.
- PFR9042**
SLTUSER and OMTUSER parameters cannot both be specified.
- PFR9048**
Cannot display graph because of missing data.

PRTJOBTRC (Print Job Trace) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

PRTJOBTRC Command syntax diagram

Purpose

The Print Job Trace (PRTJOBTRC) command produces performance-oriented reports used to analyze job trace data collected with the Start Job Trace (STRJOBTRC) and End Job Trace (ENDJOBTRC) commands.

Optional Parameters

MBR Specifies the name of the member in file QAPTTRCJ in which the trace data was saved by the End Job Trace (ENDJOBTRC) command.

QAJOBTRC: The standard member name, QAJOBTRC, was used.

member-name: Specify the name of the member in which the data was saved.

LIB Specifies the library in which the job trace data is saved by the ENDJOBTRC command.

QPFRDATA: The trace data was saved in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library where the trace data was saved.

RPTTYPE

Specifies the type of reports to be produced.

Note: If summary reports are selected (by specifying *BOTH or *SUMMARY for this parameter), the summary reports contain information only when transaction ending program and transaction starting program pairs are found in the collected data.

***BOTH:** Both the detail and summary reports are produced (three reports total).

***DETAIL:** A report is produced that details the individual job trace records. The output is directed to the printer file QPPTTRCD. Each page heading includes the text 'Job Trace Information'.

***SUMMARY:** Two reports are produced that summarize the job trace data by transaction. One report shows primarily physical disk activity; its printer file is QPPTTRC1, and its page heading includes the text 'Trace Analysis Summary'. The other report concentrates on higher level activities, such as database I/O and inter-program transfers of control; its printer file is QPPTTRC2, and its page heading includes the text 'Trace Analysis I/O Summary'.

TITLE Specifies a title that is printed on the page heading of each report.

***BLANK:** No title is specified.

'report-title': Specify a title of up to 50 characters, enclosed in apostrophes. This may be used, for example, to distinguish between reports on different sets of trace data or different sections of the same data.

STRSEQ

Specifies the sequence number of the first job trace record to be included in any reports. No records before this one are listed in the detail report or counted in either summary report.

***FIRST:** Trace records starting from the first one (sequence number 1) are included.

sequence-number: Specify the sequence of the first trace record to be included. A value can be determined by previewing reports produced from all the job trace data. This can be used to bracket a particular set of transactions on which to report.

ENDSEQ

Specifies the sequence number of the last job trace record to be included in any reports. No records following this one are listed in the detail report or counted in either summary report.

***LAST:** Trace records through the last one are included.

sequence-number: Specify the sequence number of the last trace record to be included. As with STRSEQ, a value can be chosen through a preview process so as to bracket a particular set of transactions.

ENDTNS

Specifies the program that signifies the end of a transaction (when followed by the STRTNS program).

QT3REQIO: The low-level OS/400 system work station I/O program, QT3REQIO, is used. This value is used to break the trace data into work station transactions.

program-name: Specify the name of the program that ends a transaction. This allows reporting on transactions not on work stations, such as communications lines.

STRTNS

Specifies the program that signifies the start of a transaction (when preceded by the ENDTNS program).

QWSGET: The OS/400 system work station input program, QWSGET, is used. This value is used to break the trace data into work station transactions.

program-name: Specify the name of the program that starts a transaction.

MODEL

Specifies the processing unit model code of the server where the job trace data was created. The model is used to adjust the summary reports to compensate for the additional processing unit time required to collect the trace data. If the wrong model is chosen, the adjusted values in the summary reports will be incorrect.

***CUR:** The model code of the processing unit, at which the current job is running, is used.

model-code: Specify the model code of the processing unit where the trace data was collected.

JOB Specifies the job name used if submitting the job for batch processing.

Note: If *NONE is specified on the JOBID parameter, this parameter is ignored; job processing is performed interactively.

PRTJOBTRC: The command name is used for the job name.

***MBR:** The name selected for the performance data member on the MBR parameter is used.

job-name: Specify the name used for batch jobs.

JOBID 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 to be searched.

QPFRJOBID: The IBM-supplied Performance Tools job description, QPFRJOBID, 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 is not available for other use during this time, which could be significant for long jobs.

Example for PRTJOBTRC

```
PRTJOBTRC LIB(MYLIB) RPTTYPE(*DETAIL)
```

This command produces a detail report using data saved in member QAJOBTRC in library MYLIB/QAPTRCJ.

Error messages for PRTJOBTRC

*ESCAPE Messages

None.

PRTLBLBRM (Print Labels Using BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed.

PRTLBLBRM Command syntax diagram

Purpose

The Print Labels using BRM (PRTLBLBRM) command prints media labels that you have selected as a result of media processing. Label information includes:

- Volume serial identifier
- Creation date
- Expiration date
- Location
- Container identifier
- Text

Note: The source for the three printer files is located in QUSRBRM/QA1ASRC. There are three members, QP1A1LP, QP1A2LP and QP1A3LP. These correspond to 6 lines per inch, 8 lines per inch, and 9 lines per inch as the options on the *Change Media Class* display.

To change the format of the printer labels, edit the source member corresponding to the labels you selected for the media. Editing can be done with an editor (for example Source Entry Utility (SEU)), but you must first give the members the correct member type of PRTF. You can do this through PDM when you are working with members. When changing the source, do not change the record name or any of the field names. The print programs depend on these named items being present. You can change the position. When you compile the printer file, be sure and specify level check (*NO) on the CRTPRTF command.

Note: If you change the printer file description, you should keep a copy. Any release upgrade may cause the printer files to change and your version may be back level.

There are no parameters for this command.

Error messages for PRTLBLBRM

None

PRTLCKRPT (Print Lock Report) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

PRTLCKRPT Command syntax diagram

Purpose

The Print Lock Report (PRTLCKRPT) command produces a report that shows lock and seize conflicts that occur during system operation. This report is produced from the resource management trace data collected by the Start Performance Trace (STRPFRTRC) command and formatted by the Print Transaction Report (PRTTNSRPT) command. This information can be used to determine whether jobs are being delayed during processing because of unsatisfied lock requests or internal machine seizes; these conditions are also known as waits.

Required Parameter

MBR Specifies the member in file QAPMDMPT in which the resource management trace data is collected by the Start Performance Trace (STRPFRTRC) command. Specify the name of the member that contains the performance data.

Optional Parameters

LIB Specifies the library where the data is saved.

QPFRDATA: The IBM-supplied performance data library, QPFRDATA, is used.

library-name: Specify the name of the library where the data is collected.

TITLE Specifies a title that prints in the heading at the top of each page of the report.

***MBRTXT:** The text of the selected member is used.

***BLANK:** No title is used.

'report-title': Specify a title of up to 50 characters enclosed in apostrophes.

RPTTYPE

Specifies the type of reports to produce.

***SUM:** The report includes only a summary of the seize/lock data.

***TOD:** The report includes detail sorted by time of day, followed by a summary.

***RQS:** The report includes detail sorted by the name of the requesting job and the time of day, followed by a summary.

***HLD:** The report includes detail sorted by the name of the holding job and the time of day, followed by a summary.

***OBJ:** The report includes detail sorted by the name of the object and the time of day, followed by a summary.

***ALL:** In addition to a summary, four reports are produced: *TOD, *RQS, *HLD, and *OBJ.

FIRST Specifies whether PRTLCKRPT has been run since the last set of resource management trace data was collected.

***YES:** This is the first time PRTLCKRPT is being run for the data. The data is reformatted and written to the member specified on the MBR parameter in QAPTLCKD; the report is created from this preprocessed data.

***NO:** PRTLCKRPT has already been run for this set of data. The preprocessing pass is not done.

PERIOD

Specifies the period of time on which to report. The value consists of two lists of one element each:

```
PERIOD((start-time) (end-time))
```

Element 1: Starting Time

Use one of the following to specify the starting time. Trace records collected prior to this time are not included in the report.

***FIRST:** Trace records starting from the earliest possible time (00:00:00) are included.

start-time: Specify the time after which data records are collected. 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: Ending Time

Use one of the following to specify the ending time. No trace records collected after this time are included in the report.

***LAST:** Trace records through the latest possible time (23:59:59) are included.

end-time: Specify the time of the last record that is included. See *start-time* in this parameter for details on how the time must be specified.

MINWAIT

Specifies the minimum wait time for a seize/lock record that is included in the report. Records with shorter wait times are not included in either the detail or summary listing.

500: The default value of 500 milliseconds (half a second) is used. Records with shorter wait times are generally of little interest when determining the source of performance problems.

number-of-milliseconds: Specify the minimum wait time, ranging from 0 through 30,000 milliseconds (from no minimum up to 30 seconds).

JOB Specifies the job name used if the job is being submitted for batch processing.

Note: If *NONE is specified on the JOBD parameter, this parameter is ignored; job processing is performed interactively.

PRTLCKRPT: The command name is used for the job name.

***MBR:** The name selected for the performance data member on the MBR parameter is used.

job-name: Specify the name used for batch jobs.

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 to be searched.

The possible job description values are:

QPFRJOBD: The IBM-supplied Performance Tools job description, QPFRJOB, is used.

job-description-name: Specify the name of an alternative 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 is not available for other use during this time, which could be significant for long jobs.

Examples for PRTLCKRPT

Example 1: Producing a Summary Report

```
PRTLCKRPT MBR(RESTRC)
```

This command produces a summary report from the performance data saved in member RESTRC of QPFRDATA/QAPMDMPT from a prior run of the Start Performance Trace (STRPFRTRC) and Print Transaction Report (PRTTNSRPT) commands.

Example 2: Including a Detail Listing Sorted By Time

```
PRTLCKRPT MBR(RESTRC) RPTTYPE(*TOD)
```

This command produces the same report as the previous example, except that it includes a detail listing sorted by the time in which the lock/seize conflicts occurred.

Error messages for PRTLCKRPT

***ESCAPE Messages**

PFR5511

Cannot access resource management trace data.

PFR5512

Cannot access processed seize or lock conflict data.

PRTMEDBRM (Print Media Exceptions Using BRM) Command

Description

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

PRTMEDBRM Command syntax diagram

Purpose

The Print Media Exceptions using BRM (PRTMEDBRM) command prints information about volumes in the BRMS/400 media inventory and designates those that have exceeded use thresholds, number of reads or writes and so on that are specified in the media class for the volume. Based on that information, you can make decisions about the reported volumes. You can specify whether to include all volumes including exceptions or exception volumes only. One of two reports can be printed based on the choice selected in the TYPE parameter. If you select *THRESHOLD, the Media Volume Threshold Information report is printed. If you select *STATISTICS, the Media Volume Statistics report is printed.

Example for PRTMEDBRM

Example 1: Printing the Volume Information report for Volumes that are Threshold Exceptions

```
PRTMEDBRM VOL(*EXCP)
```

In the example, the Volume Information report would include only volumes that have exceeded their threshold values.

Error messages for PRTMEDBRM

None

PRTMOVBRM (Print Media Movement Using BRM) Command

Description

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

>PRTMOVBRM Command syntax diagram

Purpose

The Print Media Movement using BRM (PRTMOVBRM) command prints the Media Movement report based on a specified date range, type of move that you request, that is, verify or not verify, and locations. The report shows all the volumes that have moved (or in the case of *NEXT, which volumes will move to a new location), the from and to locations, the move policy for each volume and the move date. The report, if printed, is written to printer file QP1APVMS.

The Media Movement report can be used to report volumes that have already moved or can be used to report the next scheduled media movement for a volume. Reporting of next scheduled move is performed by selecting the *NEXT variable for the TYPE parameter.

Example for PRTMOVBRM

Example 1: Printing the Media Movement Report

PRTMOVBRM TYPE(*VFY) LOC(*HOME)

In the example, media entries that have been verified (*VFY) and that are moving from the home location (*HOME) will be included in the report.

Error messages for PRTMOVBRM

None

PRTPEXRPT (Print Performance Explorer Report) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

PRTPEXRPT Command syntax diagram

Purpose

The Print Performance Explorer Report (PRTPEXRPT) command prints a formatted listing of the data that was collected by the performance explorer and saved across a set of physical files in a particular library.

Restrictions:

1. This command is shipped with PUBLIC *EXCLUDE authority.
2. The user must have read authority for each performance explorer database file in the specified library.
3. 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.

Required Parameter

MBR Specifies where the data is located for the report. This is the value that was specified for the SSNID or DTAMBR parameter when the data was saved using the End Performance Explorer (ENDPEX) command. Each database file used by the performance explorer when it saved the collected performance data should have a member with the name specified.

member-name: Specify the member name.

Optional Parameters

LIB Specifies the library where the data will be found.

QPEXDATA: The collected data exists in database files in library QPEXDATA.

library: Specify the library name which contains the database files that hold the collected performance data.

TYPE Specifies the type of report to produce. The type of report requested must match the type of data that was collected. If there is a mismatch, an error message is issued. The type of performance data collected is determined by the performance explorer definition that was specified on the Start Performance Explorer (STRPEX) command. See ADDPEXDFN for more information.

Note: An exception to the matching of types occurs when you collect data with a definition of TYPE(*TRACE) INTERVAL(nn) BASEVT(*PMCO). When you collect this trace data, you are allowed to specify a report type of *PROFILE. This type of report is known as a *TRACE collection and a *PROFILE report.

***STATS:** A statistics report is produced.

Note: This parameter is valid only for data collected by *STATS mode definitions.

***TRACE:** A trace report is produced.

Note: This parameter is valid only for data collected by *TRACE mode definitions or *PROFILE PRFTYPE(*JOB) definitions.

***PROFILE:** A profile report is produced.

Note: This parameter is valid only for data collected by *PROFILE mode definitions or *TRACE TRCTYPE(*PROFILE) definitions.

***BASIC:** A basic report is produced that includes the definition, run, and task information sections. This parameter is valid for data collected by any definition.

OUTPUT

Specifies whether the output from the command is printed with the job's spooled output or directed to a database file.

Note: This parameter is valid only if TYPE(*TRACE) is specified.

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

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

OUTFILE

Specifies the name of the database file to which the output of the command is directed. If this file does not exist, this command creates a database file in the specified library. The public authority is the same as the create authority specified for the library in which the file is created.

Notes:

1. The file specified here cannot be a DDM file.
2. The model file QAVPETRCI resides in library QPFR.

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

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

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

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

OUTMBR

Specifies the name of the database file member to which the output is directed.

Element 1: Member to Receive Output

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

member-name: Specify the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it. If the member already exists, you have the option to add new records to the end of the existing member or clear the member and then add the new records.

Element 2: Operation to Perform on Member

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

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

TRACEOPT

Specifies how to organize a trace (*TRACE) report. Records are ordered based on the value specified for the ORDER parameter.

Element 1: Sort by

This value represents how the data is ordered.

***TIMESTAMP**: The records are listed in time stamp order.

***TASK**: The records are listed in time stamp order within each job or task.

Element 2: Omit completion records

This provides a mechanism to reduce large amounts of data to enable more efficient review of the data.

***NO**: All records associated with this performance data collection session are included in the report.

***YES**: All completion records are excluded from the report. This is helpful if there is a large amount of data to review.

Element 3 Omit category

Specify one or more categories to be omitted from the generated report.

The possible single value is:

***NONE**: No categories are omitted.

The possible omitted category values are:

***PGM**: Exclude the category for the program call flow events.

***LICPGM**: Exclude the category for the Licensed Internal Code call flow events.

***ASM**: Exclude the category for the auxiliary storage management events.

***BASE**: Exclude the category for the base events, which includes tasking events.

***DISK**: Exclude the category for the direct access storage device events.

***DSKSVR**: Exclude the category for the disk server events.

***FAULT**: Exclude the category for the page fault events.

***JOB**: Exclude the category for the job or process management events.

***LOCK**: Exclude the category for the seize lock events.

***SAR**: Exclude the category for the segment address register events.

***MIBRKT**: Exclude the category for the machine interface program bracketing events.

***LICBRKT**: Exclude the category for the Licensed Internal Code bracketing events.

***DASD**: Exclude the category for the direct access storage device events.

***DASDSVR:** Exclude the category for the DASD server events.

***PAGEFLT:** Exclude the category for the page fault events.

***RMPPR:** Exclude the category for the resource management process management events.

***RMSZ:** Exclude the category for the resource management seize lock events.

TRCTYPE

Specifies which trace events to include in the output. The options possible are the same options found on the ADDPEXDFN command.

The possible single value is:

***ALL:** Include all trace events in the output.

The possible trace type values are:

***CALLRTN:** Specifies that call return events are included in the output. Call return events occur when a program is entered and exited as well as when certain machine instructions are started and completed.

***BASIC:** Specifies that events relative to general performance analysis are included in the output.

***DSKIO1:** Specifies that events associated with disk input/output operations are included in the output.

***DSKIO2:** Specifies that events associated with the disk input/output operations plus higher level requests to do input/output operations are included in the output.

***DSKSVR:** Specifies that events associated with disk server operations are included in the output.

***DSKSTG:** Specifies that events associated with disk storage consumption are included in the output.

***VRTADR:** Specifies that events associated with virtual address assignment are included in the output.

***PGMACT:** Specifies that events associated with program activations and deactivations are included in the output.

***FILEOPEN:** Specifies that events associated with file opens are included in the output.

***PFRDTA:** Specifies that events associated with CPU instruction profiling are included in the output. The *PFRDTA value provides you with a detailed list of files. To receive a list in a summary format, as an alternative, you can specify PRTPEXRPT TYPE(*PROFILE).

***TASKSWT:** Specifies that events associated with tasking are included in the output.

PERIOD

Specifies the period of time on which to report. The parameter consists of two lists of two elements each. Data collected prior to the starting time on the starting date and after the ending time on the ending date is not included in the report.

Element 1: Starting time

***AVAIL:** The recorded data that is available for the specified starting date is shown.

start-time: Specify the starting time on the specified starting date that indicates the recorded data to be shown. The time is specified in 24-hour format with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from

the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

All time and date entries must be 2 digits in length. This means that you must include zeros.

Element 2: Starting date

***CURRENT:** The recorded data for the current day and between the specified starting and ending times (if specified) is shown.

***BEGIN:** The recorded data from the beginning of the log is shown.

start-date: Specify the date printed. The date must be entered in the format specified by the system value QDATFMT, and if separators are used, as specified by the system value QDATSEP.

Element 3: Ending time

***AVAIL:** The recorded data that is available for the specified ending date is shown.

end-time: Specify the ending time for the specified ending date that determines the recorded date that is printed.

Element 4: Ending date

***CURRENT:** The current day is the last day for which recorded data is shown.

***END:** The last day on which data was logged is shown. If PERIOD(*END) is specified, a time value other than *AVAIL for end time is ignored.

end-date: Specify the ending date for which recorded data is to be printed. The date must be entered in the format specified by the system value QDATFMT, and if separators are used, as specified by the system value QDATSEP.

SLTJOB

Specifies which jobs to include from the report. This allows you to narrow the scope of the performance explorer report by selecting specific jobs.

Note: The SLTJOB and OMTJOB parameters are mutually exclusive.

Element 1: Job name

***ALL:** All jobs in the performance explorer database are included.

job-name: Specify the name of the job to be included in the performance explorer report.

generic-job-name:* Specify the generic name of the job to be included in the performance explorer report. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

Element 2: User name

***ALL:** All jobs that match the specified job name are included.

user-name: Specify the name of the user of the job to be included.

generic-user-name:* Specify the generic name of the user of the jobs to be included. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

Element 3: Job number

***ALL:** All jobs that match the specified job name and user name are included.

job-number: Specify the job number to further qualify the job name and user name.

OMTJOB

Specifies which jobs are omitted from the report. This allows you to narrow the scope of the performance explorer report by omitting specific jobs.

Note: The SLTJOB and OMTJOB parameters are mutually exclusive.

Element 1: Job name

***NONE:** No jobs in the performance explorer database are omitted.

job-name: Specify the name of the job to be omitted in the performance explorer report.

generic-job-name:* Specify the generic name of the job to be omitted in the performance explorer report. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

Element 2: User name

***ALL:** All jobs that match the specified job name will be omitted.

user-name: Specify the name of the user of the job to be omitted.

generic-user-name:* Specify the generic user name of the jobs to be omitted. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

Element 3: Job number

***ALL::** All jobs that match the specified job name and user name will be omitted.

job-number: Specify the job number to further qualify the job name and user name.

STATSOPT

Specifies how to organize a statistics (*STATS) report. Records are ordered based on the value specified for the ORDER parameter.

Note: This parameter is ignored if, on the ADDPEXDFN command, you specified TYPE(*STATS) and DTAORG(*HIER). The parameter is ignored to retain the parent-child relationship that was collected for this definition.

Element 1: Sort by

Specifies how the records are arranged in the report.

***CPU:** Arrange the output by amount of CPU time.

***PGMNAME:** Arrange the output by program name.

***INVCNT:** Arrange the output by number of times a program or a procedure is called.

***DBSYNCIO:** Arrange the output by amount of database related synchronous I/O.

- ***DBASYNCIO**: Arrange the output by amount of database related asynchronous I/O.
- ***NDBSYNCIO**: Arrange the output by amount of non-database related synchronous I/O.
- ***NDBASYNCIO**: Arrange the output by amount of non-database related asynchronous I/O.
- ***MICALLS**: Arrange the output by number of MI calls.
- ***MIINST**: Arrange the output by MI instruction name.
- ***CUMLCPU**: Arrange the output by cumulative CPU value.
- ***CUMLDBSYNCIO**: Arrange the output by cumulative amount of database synchronous I/O.
- ***CUMLDBASYNCIO**: Arrange the output by cumulative amount of database asynchronous I/O.
- ***CUMLNDBSYNCIO**: Arrange the output by cumulative amount of non-database synchronous I/O.
- ***CUMLNDBASYNCIO**: Arrange the output by cumulative amount of non-database asynchronous I/O

Element 2: Summarize by

- ***PROGRAM**: The data is summarized at the program level.
- ***BLANK**: The data is not summarized.
- ***MODULE**: The data is summarized at the module level.

PROFILEOPT

Specifies how to organize a profile (*PROFILE) report. Records are ordered based on the value specified for the ORDER parameter.

Element 1: Sort by

Specifies how the records are arranged in the report.

- ***SAMPLECOUNT**: Arrange the output relative to the sample count.
- ***ADDRESS**: Arrange the output relative to the sampled address.

Element 2: Summarize By

- ***PROGRAM**: Summarize the data at the program level.
- ***STATEMENT**: Summarize the data at the statement level.
- ***PROCEDURE**: Summarize the data at the procedure level.
- ***MODULE**: Summarize the data at the module level.
- ***BLANK**: No summary records are provided.

Element 3: Filter percentage

This provides a filter to eliminate the insignificant records. For example, an entry of 10 would omit all the records that contain less than 10% of the samples taken during the collection.

0: No records are omitted from the report.

filter-percentage: Specify a number in the range of 0 to 100.

ORDER

Specifies how the data should be ordered in the report.

***DESCENDING**: The data records are ordered in descending order. If records are sorted by a numeric field, records are ordered from largest to smallest. If records are sorted by a name field, records are in reverse alphabetical order, for example, from Z to A.

***ASCENDING:** The data records are ordered in ascending order. If records are sorted by a numeric field, records are ordered from smallest to largest. If records are sorted by a name field, records are in alphabetical order, for example, from A to Z.

NBRTHD

Specifies the number of concurrent threads that the PRTPEXRPT command uses to print the data. Specifying a number greater than 1 allows the PRTPEXRPT command to take advantage of available CPU cycles, especially on a multi-processor system. While this may speed up the command processing, it may also degrade the performance of other jobs on the system. You can minimize this impact by changing the priority of the job that runs the PRTPEXRPT command to a higher number. You should also verify that the disk subsystem can handle the additional threads. Typically, the PRTPEXRPT command requires one disk arm for each active thread.

***CALC:** The system calculates a reasonable number of threads to do the command processing which does not use excessive system resources. Usually this is one or two threads for each available processor. If this command is run in an interactive job, *CALC uses only one thread.

number-of-threads: Specify the number of threads for the PRTPEXRPT command to use to process the collected data.

Examples for PRTPEXRPT

Example 1: Statistics Report

```
PRTPEXRPT MBR(SAMPLE) LIBRARY(SAMPLELIB)
TYPE(*STATS) STATSOPT(*INVCNT *MODULE)
```

In this example, a statistics type report is generated based on data members named SAMPLE in library SAMPLELIB. The data is arranged in descending order based on invocation counts and is summarized at the module level.

Example 2: Profile Report

```
PRTPEXRPT MBR(SAMPLE2) TYPE(*PROFILE)
PROFILEOPT(*SAMPLECOUNT *PROGRAM)
ORDER(*DESCENDING)
```

In this example, a profile type report is generated based on data members named SAMPLE2 in the default library, QPEXDATA. The data is arranged in descending order based on the sample count and is summarized at the program level.

Error messages for PRTPEXRPT

None.

PRTTCPPTP (Print Point-to-Point TCP/IP Profile) Command Description

PRTTCPPTP Command syntax diagram

Purpose

The Print Point-to-Point TCP/IP Profile (PRTTCPPTP) command is used to print the configuration data for a point-to-point TCP/IP profile. Printer file QPTOCPPP is used to generate the spooled file. The spooled file name will be the same as the point-to-point profile name, and the spooled file user data will be 'PRTTCPPTP'.

Required Parameter

CFGPRF

Specifies the name of the point-to-point configuration profile to print.

configuration-profile-name: Specify the name of a valid configuration profile.

Example for PRTTCPPTP

```
PRTTCPPTP CFGPRF(ANSPROFILE)
```

This command prints the configuration data for point-to-point profile ANSPROFILE. The spooled file name will be ANSPROFILE and the spooled file user data will be 'PRTTCPPTP'.

Error messages for PRTTCPPTP

*ESCAPE Messages

TCP83F1

Point-to-point profile &1 not printed.

PRTPOLRPT (Print Pool Report) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

PRTPOLRPT Command syntax diagram

Purpose

The Print Pool Report (PRTPOLRPT) command produces a pool-oriented report from the performance data collected by Collection Services. The report is written to the printer file QPPTITVP. The two sections of the report are subsystem activity and workload activity of storage pools. The information is presented according to interval order. Jobs may be selectively included in, or excluded from, the report based on a variety of job details and interval times.

Required Parameter

MBR Specifies the performance data member that is used. This name should correspond to the member name specified on the TOMBR parameter of the Create Performance Data (CRTPFRDTA) command.

Optional Parameters

LIB Specifies the library where the performance data is located.

QPFRDATA: The performance data files are located in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library where the performance data is located.

TITLE Specifies a title that is printed in the heading at the top of each page of the report.

***MBRTXT:** The text of the database member, which contains the performance data, is the report title.

***BLANK:** A blank title is used.

'report-title': Specify a title of up to 50 characters, enclosed in apostrophes.

PERIOD

Specifies the period of time on which to report. The parameter consists of four elements: a starting

time and date, and an ending time and date. Data collected prior to the starting time on the starting date and after the ending time on the ending date is not included in the report.

```
PERIOD((start-time start-date)
       (end-time end-date))
```

*N may be used to maintain the position in the parameter value sequence in place of an element that precedes the values that are specified. For example, PERIOD(*N (*N 091290)) specifies the ending date and uses the defaults for the other values.

Element 1: Starting Time

One of the following values is used to specify the starting time.

***FIRST:** Data records starting from the beginning of the first day (00:00:00) of the collection period are included.

start-time: Specify the time of the first data record to include in the report. 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.

All time and date entries must be two digits in length. This means that you must include zeros.

Element 2: Starting Date

One of the following values is used to specify the starting date. Data collected prior to the starting time on this date is not included in the report.

***FIRST:** Data records starting from the first day of the collection period are included in the report.

start-date: Specify the date of the first data record to include in the report. The date must be entered in the format specified by QDATFMT and, if separators are used, QDATSEP. For instance, the system might have a date format of 'mm/dd/yy'. The month (mm), day (dd), and year (yy) are all required 1- or 2-digit values. The slashes (/) are optional if all six digits are specified. If the slashes are omitted, or if the value is entered from the prompt display, the apostrophes (') are not required.

Element 3: Ending Time

One of the following values is used to specify the ending time. Data collected after this time on the ending date is not included in the report.

***LAST:** Data records through the end of the day (23:59:59) are included in the report.

end-time: Specify the time of the last data record to include in the report. See *start-time* in this parameter for details on how the time must be specified.

Element 4: Ending Date

One of the following values is used to specify the ending date. Data collected after the ending time in this date is not included in the report.

***LAST:** Data records through last day of the collection period are included in the report.

end-date: Specify the date of the last data record to include in the report. Use the same date format used for the starting date.

Other Single Values

***SELECT:** An interval selection display is shown from which you can select one or more intervals to include. This value is valid only in the interactive environment. If this value is used, the remaining values of this parameter (starting time and date and ending time and date) are ignored.

SLTJOB

Specifies a list of up to 50 jobs to select. Only specified jobs are included in the report.

Individual jobs are identified by a job identifier. A job identifier is either the special value *NONE or a qualified name with up to three elements: a job number, a user name, and a job name. Job identifiers are written in the form: job-number/user-name/job-name, but you do not have to specify all three elements. *N can be used in place of an element to maintain the position in the parameter value sequence.

A job identifier is either the special value *ALL or a qualified name with up to three elements, for example:

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

Note: The SLTJOB and OMTJOB parameters are mutually exclusive.

Element 1: Job name

***ALL:** All jobs are included, unless excluded by some other selection criterion.

job-name: Specify the names of the jobs to select. Because jobs may have identical job names, this value may not identify a specific job. You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

user-name: Specify the user name of the job to select. Because jobs may have identical user names, this value may not identify a specific job.

job-number: Specify the 6-digit number of the job to select. All six digits must be specified; use leading zeros if necessary.

Element 2: Thread

***ALL:** All threads are included, unless excluded by some other selection criterion.

thread-identifier: Specify the thread identifier to select. Because some jobs may have identical thread identifiers, this value may not identify a specific job.

OMTJOB

Specifies a list of up to 50 jobs to omit. All jobs specified are excluded from the report.

Individual jobs are identified by a job identifier. A job identifier is either the special value *NONE or a qualified name with up to three elements: a job number, a user name, and a job name. Job identifiers are written in the form: job-number/user-name/job-name, but you do not have to specify all three elements. *N can be used in place of an element to maintain the position in the parameter value sequence.

Note: The SLTJOB and OMTJOB parameters are mutually exclusive.

Element 1: Job name

***NONE:** Jobs are not excluded based on job identifier.

job-name: Specify the names of the jobs to omit. Because jobs may have identical job names, this value may not identify a specific job. You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

user-name: Specify the user name of the job to omit. Because jobs may have identical user names, this value may not identify a specific job.

job-number: Specify the 6-digit number of the job to omit. All six digits must be specified; use leading zeros if necessary.

Element 2: Thread

***ALL:** All threads are included, unless excluded by some other selection criterion.

thread-identifier: Specify the thread identifier to select. Because some jobs may have identical thread identifiers, this value may not identify a specific job.

SLTUSRID

Specifies a list of up to 50 user names to be included in the report. Only jobs with one of the specified user names are included in the report.

Note: SLTUSRID and OMTUSRID parameters are mutually exclusive.

***ALL:** Jobs with all user names are included, unless excluded by some selection criterion.

user-name: Specify the user names of the jobs to select. Because jobs may have identical user names, this value may not identify a specific job. SLTUSRID(user) is equivalent to SLTJOB(*N/user/*N). You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

OMTUSRID

Specifies a list of user names to omit. Jobs with any of the user names specified are excluded from the report.

***NONE:** Jobs are not excluded based on user name.

user-name: Specify the specific or generic user name of the job to omit. Because jobs may have identical user names, this value may not identify a specific job. OMTUSRID(user) is equivalent to OMTJOB(*N/user/*N).

SLTPOOLS

Specifies a list of up to 64 pools to be included in the report. Only jobs that run in one of the specified pools are included in the report.

Note: The SLTPOOLS and OMTPOOLS parameters are mutually exclusive.

***ALL:** Jobs in all pools are included, unless excluded by other selection criteria.

storage-pool-identifier: Specify the number of a pool to select. Valid values range from 1 through 64.

OMTPOOLS

Specifies a list of up to 64 pools to omit. Jobs that run in any of the specified pools are excluded from the report.

Note: The SLTPOOLS and OMTPOOLS parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on their pool.

storage-pool-identifier: Specify the number of the pool to omit. Valid values range from 1 through 64.

SLTSBS

Specifies a list of up to 50 subsystems to select. Only jobs that run in one of the specified subsystems are included in the report.

Note: The SLTSBS and OMTSBS parameters are mutually exclusive.

***ALL:** Jobs in all subsystems are included unless excluded by other selection criteria.

subsystem-name: Specify the name of a subsystem to select.

OMTSBS

Specifies a list of up to 50 subsystems to omit. Jobs that run in any of the specified subsystems are excluded from the report.

Note: The SLTSBS and OMTSBS parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on subsystem.

subsystem-name: Specify the name of a subsystem to omit.

SLTLINE

Specifies a list of up to 50 communications lines to select. Only jobs that use a remote device connected through one of the specified communications lines are included in the report.

Note: The SLTLINE and OMTLINE parameters are mutually exclusive.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

communications-line-name: Specify the name of a communications line to select. This excludes jobs using remote devices connected through other communications lines (or no communications line), even if the controllers to which these devices are attached are specified on the SLTCTL parameter.

OMTLNE

Specifies a list of up to 50 communications lines to omit. Jobs that use a remote device connected through any of the specified communications lines are excluded from the report.

Note: The SLTLNE and OMTLINE parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on communications line.

communications-line-name: Specify the name of a communications line to omit.

SLTCTL

Specifies a list of up to 50 communications controllers to select. Only jobs that use a device connected to one of the specified communications controllers are included in the report.

Note: The SLTCTL and OMTCTL parameters are mutually exclusive.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

controller-name: Specify the name of a communications controller to select.

OMTCTL

Specifies a list of up to 50 communications controllers to omit. Jobs that use a device connected to any of the specified communications controllers are excluded from the report.


Note: The SLTCTL and OMTCTL parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on communications controllers.

controller-name: Specify the name of a communications controller to omit.

SLTFCNARA

Specifies a list of functional areas to select. Only jobs and users identified in one of the functional areas are included in the report.


A functional area is a list of job names and user names previously defined by the user. Refer to the Performance Tools for iSeries  book for more information on functional areas.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

functional-area-name: Specify the name of the functional area to select.

OMTFCNARA

Specifies a list of functional areas to omit. Jobs and users identified in any of the functional areas are excluded from the report.

A functional area is a list of job names and user names previously defined by the user. Refer to the Performance Tools for iSeries  book for more information on functional areas.

***NONE:** Jobs are not excluded based on functional area.

functional-area-name: Specify the name of the functional area to omit.

JOB

Specifies the job name used if the job is submitted for batch processing.

Note: If *NONE is specified on the JOBD parameter, this parameter is ignored; job processing is performed interactively.

PRTPOLRPT: The command name is used for the job name.

***MBR**: The name selected for the performance data member on the MBR parameter is used.

job-name: Specify the name used for batch jobs.

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 to be searched.

QPFRJOBD: The IBM-supplied Performance Tools job description, QPFRJOB, is used.

job-description-name: Specify the name of an alternate job description.

***NONE**: A batch job is not submitted; instead, processing continues interactively while the user waits. The user's work station is not available for other use during this time, which could be significant for long jobs.

Examples for PRTPOLRPT

Example 1: Printing a Report

```
PRTPOLRPT MBR(DTA071588A)
```

This command submits a batch job to print a report on all jobs in all intervals in the member DTA071588A of the performance data files in library QPFRDATA. The report title is taken from the text of that member.

Example 2: Selecting Intervals to Include in Report

```
PRTPOLRPT MBR(DTA071588A) PERIOD(*SELECT)
```

This command submits a job to print a report from the same data, but first shows a display from which the user interactively selects the intervals to include.

Example 3: Specifying Data Collection Time Period

```
PRTPOLRPT MBR(DTA071588A) PERIOD((2330)(0130))
```

This command submits a job to print a report on the data collected from 11:30 PM on the first day of collection through 1:30 AM on the last day of collection. However, if data collection started and ended on the same day, an error message is printed, because the specified ending date and time is before the specified starting date and time.

Example 4: Specifying a User ID

```
PRTPOLRPT MBR(DTA071588A) SLTUSRID(D46*)  
JOB(*NONE)
```

This command interactively prints a report for all jobs with a user ID starting with D46.

Example 5: Specifying a User ID

```
PRTPOLRPT MBR(DTA071588A) SLTJOB(D46*/*N)  
JOB(*NONE)
```

This command performs the same function as the previous example.

Error messages for P RTPOLRPT

***ESCAPE Messages**

PCY0013

Performance data files are not upward compatible.

PCY0014

Performance data files are not downward compatible.

PFR1010

Cannot process request because of missing data.

PFR3002

Cannot print report because of missing data.

PFR3004

Incorrect measurement interval specified.

PFR3006

Measurement interval specified is not valid.

PFR3101

The SLTJOB and OMTJOB parameters are mutually exclusive.

PFR3102

The SLTUSRID and OMTUSRID parameters cannot both be specified.

PFR3103

The SLTPOOLS and OMTPOOLS parameters cannot both be specified.

PFR3104

The SLTSBS and OMTSBS parameters cannot both be specified.

PFR3105

The SLTLINE and OMTLIE parameters cannot both be specified.

PFR3106

SLTCTL and OMTCTL parameters cannot both be specified.

PFR3107

SLTFCNARA and OMTFCNARA parameters cannot both be specified.

PFR3108

SLTLOC and OMTLOC parameters cannot both be specified.

PFR3111

Functional area &1 does not exist.

PFR5501

Performance data files are not upward compatible.

PFR5502

Performance data files are not downward compatible.

PFR9005

YAXIS(*TIME) must be specified.

PFR9042

SLTUSER and OMTUSER parameters cannot both be specified.

PFR9048

Cannot display graph because of missing data.

PRTPVTAUT (Print Private Authorities) Command Description

PRTPVTAUT Command syntax diagram

Purpose

The Print Private Authorities (PRTPVTAUT) command allows you to print a report of all the private authorities for objects of a specified type in a specified library, folder, or directory. The report lists all objects of the specified type and the users that are authorized to the object. This is a way to check for different sources of authority to objects.

This command prints three reports for the selected objects. The first report (Full Report) contains all of the private authorities for each of the selected objects.

The second report (Changed Report) contains additions and changes to the private authorities to the selected objects if the PRTPVTAUT command was previously run for the specified objects in the specified library, folder, or directory. Any new objects of the selected type, new authorities to existing objects, or changes to existing authorities to the existing objects are listed in the 'Changed Report'. If the PRTPVTAUT command was not previously run for the specified objects in the specified library, folder, or directory, there will be no 'Changed Report'. If the command has been previously run but no changes have been made to the authorities on the objects, then the 'Changed Report' is printed but there are no objects listed.

The third report (Deleted Report) contains any deletions of privately authorized users from the specified objects since the PRTPVTAUT command was previously run. Any objects that were deleted or any users that were removed as privately authorized users are listed in the 'Deleted Report'. If the PRTPVTAUT command was not previously run, there will be no 'Deleted Report'. If the command has been previously run but no delete operations have been done to the objects, then the 'Deleted Report' is printed but there are no objects listed.

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

Required Parameter

OBJTYPE

Specifies the type of object to search for. For a complete list of object types, position the cursor on the field for the Object type prompt (OBJTYPE parameter), and press F4.

object-type: Specify the object type to be processed.

Optional Parameters

CHGRPTONLY

Specifies whether just the changed reports should be printed.

***NO:** The full and changed reports are printed.

***YES:** Only the changed report and deleted report are printed.

LIB Specifies the name of the library to search for objects to be included in the private authority report.

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

This is a required parameter for all object types except *AUTL, *BLKSF, *CFGL, *CNL, *COSD, *CTL, *DEVD, *DIR, *DOC, *FLR, *LIB, *LIND, *MODD, *NWID, *NWSD, *OOPOOL, *SOCKET, *STMF, *SYMLNK, and *USRPRF.

AUTTYPE

Specifies whether object level authority, field level authority, or both object level and field level authority reports are generated. Field level authority information only applies to *FILE objects.

***OBJECT:** Object level authority reports are generated for the specified objects.

***FIELD:** For each data base file that has field level authorities, a field level authority report is generated.

This value is only valid if *FILE is specified for the Object type prompt (OBJTYPE parameter).

***ALL:** For each data base file that has field level authorities, a field level authority report is generated. Also, the object level authority reports for all the files in the specified library are generated.

This value is only valid if *FILE is specified for the Object type prompt (OBJTYPE parameter).

FLR Specifies the name of the folder to search for documents to be included in the private authority report.

This is a required parameter if *DOC is specified for the Object type prompt (OBJTYPE parameter).

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

AUTLOBJ

Specifies whether the Display Authorization List Objects (DSPAUTLOBJ) command will be run for each of the authorization lists on the system. DSPAUTLOBJ provides a list of all the objects that are secured by a specific authorization list. This parameter is only used if the object type is *AUTL. It is ignored for all other object types.

***NO:** The DSPAUTLOBJ command will not be run for each of the authorization lists on the system.

***YES:** The DSPAUTLOBJ command will be run for each of the authorization lists on the system. The output for the command will be sent to the same output queue as the authorization list report.

DIR Specifies the name of the directory to search for objects to be included in the private authority report. Only local objects in the Root, QOpenSys, and User-Defined file systems are supported.

This is a required parameter if *BLKSF, *DIR, *OOPOOL, *SOCKET, *STMF, or *SYMLNK is specified for the OBJTYPE parameter.

'directory-name': Specify the name of the directory to be searched.

SCHSUBDIR

Specifies whether to search the subdirectories for objects to be included in the private authority report. This parameter is used only when the OBJTYPE parameter is *BLKSF, *OOPOOL, *SOCKET, *STMF, or *SYMLNK.

***NO:** The subdirectories are not searched.

***YES:** The subdirectories are searched.

Example for PRTPVTAUT

```
PRTPVTAUT OBJTYPE(*FILE) LIB(PAYROLLLIB)
```

This command creates the full, changed, and deleted reports for all file objects in the library PAYROLLLIB.

Error messages for PRTPVTAUT

***ESCAPE Messages**

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

PRTPRFINT (Print Profile Internals) Command Description

PRTPRFINT Command syntax diagram

Purpose

The Print Profile Internals (PRTPRFINT) command allows you to print a report containing percentages indicating how much of the user profile object (*USRPRF) has been used. In other words, you can print a report that describes how *full* the user profile object is.

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

Optional Parameters

SELECT

Specifies what criteria is used to select the user profiles to include in the report.

***USRPRF:** User profiles are included in the report based on the value specified for the USRPRF parameter.

***PCTFULL:** User profiles are included in the report based on the value specified for the PCTFULL parameter.

USRPRF

If *USRPRF was specified for the SELECT parameter, the report will be printed for the specified user profiles.

***ALL:** The report will be run for all user profiles.

generic-user-profile-name:* Specify the generic name for the user profiles for which the report is to be run.

user-profile-name: Specify the name of the user profile for which the report is to be run.

PCTFULL

User profiles that are at least as full as the percentage specified on this parameter will be included in the report. The value specified must be between 0.01 and 100.00.

99.90: User profiles that are at least 99.9 percent filled with entries will be included in the report.

percent-full: Specify a value, ranging from 0.01 through 100.00, for the percent full selection value.

Example for PRTPRFINT

```
PRTPRFINT  SELECT(*PCTFULL) PCTFULL(99.00)
```

This report prints user profile internal information for all of the user profiles that are at least 99 percent full.

Error messages for PRTPRFINT

*ESCAPE Messages

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

PRTPUBAUT (Print Publicly Authorized Objects) Command Description

PRTPUBAUT Command syntax diagram

Purpose

The Print Publicly Authorized Objects (PRTPUBAUT) command allows you to print a report of the specified objects that do not have public authority of *EXCLUDE. For *PGM objects, only the programs that do not have public authority of *EXCLUDE that a user can call (the program is either user domain or the system security level (QSECURITY system value) is 30 or below) will be included in the report. This is a way to check for objects that every user on the system is authorized to access.

This command will print two reports. The first report (Full Report) will contain all of the specified objects that do not have public authority of *EXCLUDE. The second report (Changed Report) will contain the objects that now do not have public authority of *EXCLUDE that did have public authority of *EXCLUDE or did not exist when the PRTPUBAUT command was previously run. If the PRTPUBAUT command was not previously run for the specified objects and library, folder, or directory, there will be no 'Changed Report'. If the command has been previously run, but no additional objects do not have public authority of *EXCLUDE, then the 'Changed Report' will be printed but there will be no objects listed.

Restrictions: You must have *ALLOBJ special authority to use this command.

Required Parameter

OBJTYPE

Specifies the type of object to search for. For a complete list of object types, position the cursor on the field for the Object type prompt (OBJTYPE parameter), and press F4.

object-type: The name of the object type to search for.

Optional Parameters

CHGRPTONLY

Specifies whether just the changed report should be printed.

***NO:** The full and changed reports will be printed.

***YES:** Only the changed report will be printed.

LIB Specifies the name of the library to search for objects with public authority that is not *EXCLUDE.

***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:** All user libraries are searched.

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

This is a required parameter for all object types except *AUTL, *BLKSF, *CFGL, *CNL, *COSD, *CTLD, *DEVD, *DIR, *DOC, *FLR, *LIB, *LIND, *MODD, *NWID, *NWSD, *OOPOOL, *SOCKET, *STMF, *SYMLNK, and *USRPRF.

FILAUT

Specifies whether the Print Publicly Authorized Objects (PRTPUBAUT) command will be run for *FILE objects for each of the libraries that do not have public authority of *EXCLUDE. This parameter is only used when the OBJTYPE is *LIB.

***NO:** The PRTPUBAUT command will not be run for *FILE objects for each of the libraries that does not have public authority of *EXCLUDE.

***YES:** The PRTPUBAUT command will be run for *FILE objects for each of the libraries that does not have public authority of *EXCLUDE.

CMDAUT

Specifies whether the Print Publicly Authorized Objects (PRTPUBAUT) command will be run for *CMD objects for each of the libraries that do not have public authority of *EXCLUDE. This parameter is only used when the OBJTYPE is *LIB.

***NO:** The PRTPUBAUT command will not be run for *CMD objects for each of the libraries that does not have public authority of *EXCLUDE.

***YES:** The PRTPUBAUT command will be run for *CMD objects for each of the libraries that does not have public authority of *EXCLUDE.

PGMAUT

Specifies whether the Print Publicly Authorized Objects (PRTPUBAUT) command will be run for *PGM objects for each of the libraries that do not have public authority of *EXCLUDE. This parameter is only used when the OBJTYPE is *LIB.

***NO:** The PRTPUBAUT command will not be run for *PGM objects for each of the libraries that does not have public authority of *EXCLUDE.

***YES:** The PRTPUBAUT command will be run for *PGM objects for each of the libraries that does not have public authority of *EXCLUDE.

JOBDAUT

Specifies whether the Print Job Description Authority (PRTJOBDAUT) command will be run for each of the libraries that does not have public authority of *EXCLUDE. The PRTJOBDAUT command will list all of the job descriptions in the library that do not have public authority of *EXCLUDE and have a user name specified. This parameter is only used when the OBJTYPE is *LIB.

***NO:** The PRTJOBDAUT command will not be run for each of the libraries that does not have public authority of *EXCLUDE.

***YES:** The PRTJOBDAUT command will be run for each of the libraries that does not have public authority of *EXCLUDE.

FLR

Specifies the name of the folder to search for documents with *PUBLIC authority that is not *EXCLUDE.

This is a required parameter if *DOC is specified for the Object type prompt (OBJTYPE parameter).

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

DIR

Specifies the pathname of the directory to search for objects that do not have public authority of *EXCLUDE. Only local objects in the Root, QOpenSys, and User-Defined file systems are supported.

This is a required parameter if *BLKSF, *DIR, *OOPOOL, *SOCKET, *STMF, or *SYMLNK is specified for the OBJTYPE parameter.

'directory-name': Specify the name of the directory to be searched.

SCHSUBDIR

Specifies whether to search the subdirectories for objects to be included in the public authority report. This parameter is used only when the OBJTYPE parameter is *BLKSF, *OOPOOL, *SOCKET, *STMF, or *SYMLNK.

***NO:** The subdirectories are not searched.

***YES:** The subdirectories are searched.

Example for PRTPUBAUT

```
PRTPUBAUT OBJTYPE(*FILE) LIB(QSYS)
```

The command will create the full and changed reports for the file objects in the library QSYS.

Error messages for PRTPUBAUT

*ESCAPE Messages

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

PRTQAUT (Print Queue Authority) Command Description

PRTQAUT Command syntax diagram

Purpose

The Print Queue Authority (PRTQAUT) command allows you to print a report of the output queue and job queue authority information for the objects in the specified library. This command provides a way to check the authority attributes of the output queue and job queue objects on the system.

This command prints two reports for a library. The first report (Full Report) contains all of the output queues and job queues in the specified library. The second report (Changed Report) contains the output queues and job queues that have been created or had the authority attributes changed since the PRTQAUT command was last run for the library. If the PRTQAUT command was not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional queue information is available then the 'Changed Report' is printed but there are no queues listed.

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

Optional Parameters

LIB Specifies the name of the library to search for output queue and job queue objects to report.

***ALL:** ➤ All libraries in the auxiliary storage pools (ASPs) associated with the current job are searched. ⬅

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

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

***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.

***ALLUSR:** >> All user libraries in the ASPs associated with the current job are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

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

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries and are also searched:

QDSNX	QS36F	QUSROND
QGPL	QUSER38	QUSRPOSGS
QGPL38	QUSRADSM	QUSRPOSSA
QMPGDATA	QUSRBRM	QUSRPYMSVR
QMQMDATA	QUSRDIRCL	QUSR RDARS
QMQMPROC	QUSRDIRDB	QUSRSYS
QPFRDATA	QUSRIJS	QUSRVI
QRCL	QUSRINFSKR	QUSR VxRxMx
QRCLnnnnn	QUSRNOTES	

Notes:

1. "nnnnn" is the number of a primary auxiliary storage pool.
2. A different library name, of the form QUSR VxRxMx, 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.

CHGRPTONLY

Specifies whether just the changed report should be printed.

***NO:** The full and changed reports are printed.

***YES:** Only the changed report is printed.

Example for PRTQAUT

```
PRTQAUT LIB(QUSRSYS)
```

This command generates the full and changed reports for the output queues and job queues in the library QUSRSYS.

Error messages for PRTQAUT

***ESCAPE Messages**

CPFB307

Command &1 in use in another job.

PRTRSCRPT (Print Resource Report) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

PRTRSCRPT Command syntax diagram

Purpose

The Print Resource Report (PRTRSCRPT) command produces a device resource usage report from the performance data collected by Collection Services. The report is written to the printer file, QPPTITVR, and shows device resource information by time interval. Resources may be selected for inclusion in, or exclusion from, the report based on interval times.

Required Parameter

MBR Specifies the performance data member that is used. This name should correspond to the member name specified on the TOMBR parameter of the Create Performance Data (CRTPFRTA) command.

Optional Parameters

LIB Specifies the library where the performance data is located.

QPFRDATA: The performance data files are located in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library where the performance database files are located.

TITLE Specifies a title that prints in the heading at the top of each page of the report.

***MBRTXT:** The text of the selected member is used.

***BLANK:** No title is printed on the report.

'report-title': Specify a title of up to 50 characters enclosed in apostrophes.

PERIOD

Specifies the period of time on which to report. The parameter consists of four elements: a starting time and date, and an ending time and date. Data collected prior to the starting time on the starting date and after the ending time on the ending date is not included in the report.

```
PERIOD((start-time start-date)
       (end-time end-date))
```

*N may be used to maintain the position in the parameter value sequence in place of an element that precedes the values that are specified. For example, PERIOD(*N (*N 091290)) specifies the ending date and uses the defaults for the other values.

Element 1: Starting Time

One of the following values is used to specify the starting time. Data collected prior to this time on the starting date is not included in the report.

***FIRST:** Data records starting from the beginning of the first day (00:00:00) of the collection period are included.

start-time: Specify the time of the first data record to include in the report. 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.

The time is in 24-hour format; for example, use '13:00' for one o'clock in the afternoon.

Element 2: Starting Date

One of the following values is used to specify the starting date. Data collected prior to the starting time on this date is not included in the report.

***FIRST**: Data records starting from the first day of the collection period are included.

start-date: Specify the date of the first data record to include in the report. The date must be entered in the format specified by QDATFMT and, if separators are used, QDATSEP. For instance, the system might have a date format of 'mm/dd/yy'. The month (mm), day (dd), and year (yy) are all required for 1- or 2-digit values. The slashes (/) are optional if all six digits are specified. If the slashes are omitted, or if the value is entered from the prompt display, the apostrophes (') are not required.

Element 3: Ending Time

One of the following values is used to specify the ending time. Data collected after this time on the ending date is not included in the report.

***LAST**: Data records through the end of the day (23:59:59) are included in the report.

end-time: Specify the time of the last data record to include in the report. See *start-time* in this parameter for details on how the time must be specified.

Element 4: Ending Date

One of the following values is used to specify the ending date. Data collected after the ending time in this date is not included in the report.

***LAST**: Data records through last day of the collection period are included in the report.

end-date: Specify the date of the last data record to include in the report. Use the same date format used for the starting date.

Other Single Value

***SELECT**: Displays an internal selection screen where intervals are selected for inclusion. This value is valid only in the interactive environment. If this value is used, the remaining values of the PERIOD parameter (start-date, end-time, end-date) are ignored.

TYPE Specifies the sections of the report that you want to print.

***ALL**: All sections of the report are printed.

- ***DISK**: Specifies that you want to print the Disk Activity section.
- ***CMN**: Specifies that you want to print the Communications section.
- ***IOP**: Specifies that you want to print the IOP Utilization section.
- ***LCLWS**: Specifies that you want to print the Local Work Stations section.
- ***RMTWS**: Specifies that you want to print the Remote work Stations section.

JOB Specifies the job name to be used if the job is submitted for batch processing.

Note: If *NONE is specified on the JOBD parameter, this parameter is ignored; job processing is performed interactively.

PRTRSCRIPT: The command name is used for the job name.

***MBR**: The name selected for the performance data member on the MBR parameter is used.

job-name: Specify the name used for batch jobs.

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 to be searched.

QPFRJOB: The IBM-supplied Performance Tools job description, QPFRJOB, is used.

job-description-name: Specify the name of an alternate job description.

Other Single Values

***NONE**: A batch job is not submitted; processing continues interactively while the user waits. The user's work station is not available for other use during this time, which could be significant for long jobs.

Examples for PRTRSCRIPT

Example 1: Printing a Report

```
PRTRSCRIPT MBR(DTA071588A)
```

This command submits a batch job to print a report on all resources in all intervals in the member DTA071588A of performance data files in library QPFRDATA. The report title is taken from the text of that member.

Example 2: Selecting Intervals to Include in Report

```
PRTRSCRIPT MBR(DTA071588A) PERIOD(*SELECT)
```

This command submits a job to print a report from the same data, but first shows a screen from which the user interactively select which intervals to include.

Example 3: Specifying Data Collection Time Period

```
PRTRSCRIPT MBR(DTA071588A) PERIOD((2330)(0130))
```

This command submits a job to print a report on the data collected from 11:30 PM on the first day of collection through 1:30 AM on the last day of collection.

Error messages for PRTRSCRIPT

***ESCAPE Messages**

PCY0013

Performance data files are not upward compatible.

PCY0014

Performance data files are not downward compatible.

PFR1010

Cannot process request because of missing data.

PFR3002

Cannot print report because of missing data.

PFR3004

Incorrect measurement interval specified.

PFR3006

Measurement interval specified is not valid.

PFR3101

The SLTJOB and OMTJOB parameters are mutually exclusive.

PFR3102

The SLTUSRID and OMTUSRID parameters cannot both be specified.

PFR3103

The SLTPOOLS and OMTPOOLS parameters cannot both be specified.

PFR3104

The SLTSBS and OMTSBS parameters cannot both be specified.

PFR3105

The SLTLINE and OMTLIE parameters cannot both be specified.

PFR3106

SLTCTL and OMTCTL parameters cannot both be specified.

PFR3107

SLTFCNARA and OMTFCNARA parameters cannot both be specified.

PFR3108

SLTLOC and OMTLOC parameters cannot both be specified.

PFR3111

Functional area &1 does not exist.

PFR5501

Performance data files are not upward compatible.

PFR5502

Performance data files are not downward compatible.

PFR9005

YAXIS(*TIME) must be specified.

PFR9042

SLTUSER and OMTUSER parameters cannot both be specified.

PFR9048

Cannot display graph because of missing data.

PRTSCDJS (Print Schedule using Job Scheduler) Command Description

Note: To use this command, you must have the 5722-JS1 (Job Scheduler for iSeries) licensed program installed.

PRTSCDJS Command syntax diagram

Purpose

The Print Schedule using Job Scheduler (PRTSCDJS) command allows you print a report based on a number of days that you specify that forecasts what jobs are to be submitted by Job Scheduler and when. You can include or exclude jobs that have been held.

Note: If you use the schedule code *MINUTES, the PRTSCDJS command can be long running.

Optional Parameters

NBRDAY

Specifies the number of days that you want to forecast jobs that are scheduled to be submitted by Job Scheduler. The valid range of days are 1 to 365 and the default is 30 days.

30: The forecast will include the next 30 days.

number-of-days: Specify the number of days that you want to include in the forecast.

PAGADV

Specifies whether you want the page to advance for each day that you specify.

***YES:** The report will be printed such that at the end of each day of forecasted jobs the paper will advance to the top of a new page.

***NO:** The report will be printed continuously with page breaks when a page is filled.

INCHLDJOB

Specifies whether you want to include jobs that have been held in the forecast report.

***NO:** The report will not include jobs that have been held.

***YES:** The report will include jobs that have been held.

Example for PRTSCDJS

Example 1: Printing a Job Schedule Report

```
PRTSCDJS NBRDAY(5) PAGEADV(*NO) INCHLDJOB(*YES)
```

In this example the Job Schedule report is printed for jobs that will run for the next 5 days. There will not be a separate page for each day and held jobs are included in the report.

Error messages for PRTSCDJS

None

PRTSWL (Print Stop Word List) Command Description

PRTSWL Command syntax diagram

Purpose

The Print Stop Word List (PRTSWL) command is used to print the words from an IBM-supplied or user-created stop word list.

Required Parameter

LANGID

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

Optional Parameters

TYPE Specifies the type of stop word list to print.

***IBM:** The stop word list is IBM-supplied.

***USER:** The stop word list is user-created.

Example for PRTSWL

```
PRTSWL LANGID(ENG) TYPE(*IBM)
```

This command prints the IBM-supplied stop word list with the language ID ENG.

Error messages for PRTSWL

*ESCAPE Messages

CPF8725

&1 type stop word list not supported for language.

CPF9899

Error occurred during processing of command.

PRTSQLINF (Print Structured Query Language Information) Command Description

PRTSQLINF Command syntax diagram

Purpose

The Print Structured Query Language Information (PRTSQLINF) command prints information about the SQL statements in a program, SQL package, service program, or job. The information includes the SQL statements, the access plans used during the running of the statement, and a list of the command parameters used to precompile the source member for the object or when SQL statements are run.

Required Parameter

OBJ Specifies either the name of the object for which you want SQL information printed or *JOB indicating that the job's SQL information is to be printed. A named object can be a program, an SQL package, or a service program.

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

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

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

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

object-name: Specify the name of the program, SQL package, or service program for which you want information printed.

***JOB:** Indicates that the SQL information for the current job is to be printed. The output will only contain information for statements which have been dynamically prepared for the job. It will not contain information for SQL statements in programs, service programs, or SQL packages used by the job.

Optional Parameter

OBJTYPE

Specifies the type of object.

***PGM:** The object is a program.

***SQLPKG:** The object is an SQL package.

***SRVPGM:** The object is a service program.

Example for PRTSQLINF

Example 1: Printing SQL Information

```
PRTSQLINF  PAYROLL
```

This command will print information about the SQL statements contained in program PAYROLL.

Error messages for PRTSQLINF

*ESCAPE Messages

SQL9011

Print of SQL information failed.

PRTSBSDAUT (Print Subsystem Description Authority) Command Description

PRTSBSDAUT Command syntax diagram

Purpose

The Print Subsystem Description Authority (PRTSBSDAUT) command allows you to print a report of the subsystem descriptions in a library that contain a default user in a subsystem entry. This command provides a way to check for subsystem descriptions that allow work to be performed on your system while running under a default user profile.

This command prints two reports for a library. The first report (Full Report) contains all of the subsystem descriptions that contain a default user in a subsystem entry. The second report (Changed Report) contains the subsystem descriptions that have been changed to contain a subsystem entry with a default user since the PRTSBSDAUT command was last run for the library. If the PRTSBSDAUT command was

not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional subsystem descriptions contain subsystem entries with a default user, then the 'Changed Report' is printed but there are no subsystem descriptions listed. Changes to user profile special authorities do not cause a 'Changed Report' to be generated.

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

Required Parameter

LIB Specifies the name of the library to search for subsystem descriptions contain a subsystem entry with a default user profile specified.

***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 auxiliary storage pools (ASPs) associated with the current job are searched. <<

***ALLUSR:** >> All user libraries in the ASPs associated with the current job are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

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

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries and are also searched:

QDSNX	QS36F	QUSROND
QGPL	QUSER38	QUSRPOSGS
QGPL38	QUSRADSM	QUSRPOSSA
QMPGDATA	QUSRBRM	QUSRPYMSVR
QMQMDATA	QUSRDIRCL	QUSRRDARS
QMQMPROC	QUSRDIRDB	QUSRSYS
QPFRDATA	QUSRIJS	QUSRVI
QRCL	QUSRINFSKR	QUSRVxRxMx
QRCLnnnnn	QUSRNOTES	

Notes:

1. "nnnnn" 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.

***ALLAVL:** All libraries in all available ASPs are searched.

***ALLUSRAVL:** All user libraries in all available ASPs are searched. Refer to *ALLUSR for a definition of user libraries. <<

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

Optional Parameter

CHGRPTONLY

Specifies whether just the changed report should be printed.

***NO:** The full and changed reports will be printed.

***YES:** Only the changed report will be printed.

Example for PRTSBSDAUT

```
PRTSBSDAUT LIB(QSYS)
```

The full and changed report for all subsystem descriptions in library QSYS will be created.

Error messages for PRTSBSDAUT

*ESCAPE Messages

CPFB307

Command &1 in use in another job.

PRTSYSINF (Print System Information) Command Description

PRTSYSINF Command syntax diagram

Purpose

The Print System Information (PRTSYSINF) command spools the following system information:

1. Current disk configuration
2. History logs
3. List of user libraries
4. List of folders
5. Current system values
6. Current network attributes
7. Current configuration lists
8. user-defined edit descriptions
9. Installed PTFs
10. Reply list Entries
11. Access path recovery times (V3R1M0 and V3R2M0 systems only)
12. Service attributes (V3R1M0 and V3R2M0 systems only)
13. Network server storage spaces information (V3R1M0 and V3R2M0 systems only)
14. Power on/off schedule

15. Hardware configuration
16. Optical device descriptions
17. Configuration descriptions
18. Remote job entry descriptions
19. SNADS configuration
20. IBM supplied subsystem descriptions
21. Installed licensed programs
22. Journaling environment information

This command has no parameters.

No error messages.

PRTSYSRPT (Print System Report) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

PRTSYSRPT Command syntax diagram

Purpose

The Print System Report (PRTSYSRPT) command generates and prints a system operation overview report from the performance data collected by Collection Services. The report is written to the printer file QPPTSYSR. The system workload, resource utilization expansion, storage pool utilization, disk utilization, communications summary, and TCP/IP summary are presented in the report.

Required Parameter

MBR Specifies the performance data member that is used. This name should correspond to the member name specified on the TOMBR parameter of the Create Performance Data (CRTPFRTA) command.

Optional Parameters

LIB Specifies the library where the performance data is located.

QPFRDATA: The performance data files are located in the the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library where the performance database files are located.

TITLE Specifies the title of the report that is created.

***MBR:** The text of the database member, which contains the performance data, is used as the report title.

'report-title': Specify the title of the report. Specify up to 50 characters, enclosed in apostrophes.

PERIOD

Specifies the period of time on which to report. The parameter consists of four elements: a starting time and date, and an ending itme and date. Data collected prior to the starting time on the starting date and after the ending time on the ending date is not included in the report.

```
PERIOD((start-time start-date)
       (end-time end-date))
```

*N can be used to maintain the position in the parameter value sequence in place of an element that precedes the values that are specified. For example, PERIOD(*N (*N 091290)) specifies the ending date and uses the defaults for the other values.

Element 1: Starting Time

One of the following values is used to specify the starting time. Data collected prior to this time on the starting date is not included in the report.

***FIRST:** Data records starting from the beginning of the day (00:00:00) are included.

start-time: Specify the time of the first data record to include in the report. 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

One of the following values is used to specify the starting date. Data collected prior to the starting time on this date is not included in the report.

***FIRST:** Data records starting from the first day of the collection period are included.

start-date: Specify the date of the first data record to include in the report. The date must be entered in the format specified by QDATFMT and, if separators are used, QDATSEP. For instance, the system might have a date format of 'mm/dd/yy'. The month (mm), day (dd), and year (yy) are all required for 1- or 2-digit values. The slashes (/) are optional if all six digits are specified. If the slashes are omitted, or if the value is entered from the prompt display, the apostrophes (') are not required.

Element 3: Ending Time One of the following values is used to specify the ending time. Data collected after this time on the ending date is not included in the report.

***LAST:** Data records through the end of the day (23:59:59) are included in the report.

end-time: Specify the time of the last data record to include in the report. See *start-time* in this parameter for details on how the time must be specified.

Element 4: Ending Date One of the following values is used to specify the ending date. Data collected after the ending time on this date is not included in the report.

***LAST:** Data records through last day of the collection period included in the report.

end-date: Specify the date of the last data record to include in the report. Use the same date format used for the starting date.

Other Single Value

***SELECT:** Displays an interval selection screen where intervals are selected for inclusion. This value is valid only in the interactive environment. If this value is used, the remaining values on the PERIOD parameter (start-date, end-time, end-date) are ignored.

TYPE Specifies which sections of the System Report to print.

***ALL:** All sections are printed.

***WORKLOAD:** Workload section is printed.

***RSC:** Resource Utilization section is printed.

***RSCEXPN:** Resource Utilization Expansion section is printed.

***POOL:** Storage Pool Utilization section is printed.

***DISK:** Disk Utilization section is printed.

***CMN:** Communications Summary section is printed. This section shows data about IOP, protocol, line utilization, active devices, transactions, response times, and bytes received/transmitted.

***TCPIP:** TCP/IP Summary section is printed. This section includes additional data for TCP/IP protocols, such as packets received and transmitted and MTU sizes.

***HTTP:** The HTTP Server Summary section is printed. This section includes statistics for HTTP Server (powered by Apache).

SLTJOB

Specifies a list of up to 50 jobs to select. Only specified jobs are included in the report.

Individual jobs are identified by a job identifier. A job identifier is either the special value *NONE or a qualified name with up to three elements: a job number, a user name, and a job name. Job identifiers are written in the form: job-number/user-name/job-name, but you do not have to specify all three elements. *N can be used in place of an element to maintain the position in the parameter value sequence.

A job identifier is either the special value *ALL or a qualified name with up to three elements, for example:

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

Note: The SLTJOB and OMTJOB parameters are mutually exclusive.

Element 1: Job name

***ALL:** All jobs in the collected data are included, unless excluded by some other selection criterion.

job-name: Specify the names of the jobs to select. Because jobs may have identical job names, this value may not identify a specific job. You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

user-name: Specify the user names of the jobs to select. Because jobs may have identical user names, this value may not identify a specific job.

job-number: Specify the 6-digit number of the job to select. All six digits must be specified; use leading zeros if necessary.

Element 2: Thread

***ALL**: All threads are included, unless excluded by some other selection criterion.

thread-identifier: Specify the thread identifier to select. Because some jobs can have identical thread identifiers, this value may not identify a specific thread.

OMTJOB

Specifies a list of up to 50 jobs to omit. All jobs specified are excluded from the report.

A job identifier is either the special value *NONE or a qualified name with up to three elements: a job number, a user name, and a job name. Job identifiers are written in the form: job-number/user-name/job-name, but you do not have to specify all three elements. *N can be used in place of an element to maintain the position in the parameter value sequence.

Note: The SLTJOB and OMTJOB parameters are mutually exclusive.

Element 1: Job name

***NONE**: Jobs are not excluded based on job identifier.

job-name: Specify the names of the jobs to omit. Because jobs may have identical job names, this value may not identify a specific job. You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

user-name: Specify the user names of the jobs to omit. Because jobs may have identical user names, this value may not identify a specific job.

job-number: Specify the 6-digit number of the job to omit. All six digits must be specified; use leading zeros if necessary.

Element 2: Thread

***ALL**: All threads are included, unless excluded by some other selection criterion.

thread-identifier: Specify the thread identifier to select. Because some jobs can have identical thread identifiers, this value may not identify a specific thread.

SLTUSRID

Specifies a list of user names to select. Only jobs with one of the specified user names are included in the report.

***ALL**: Jobs with all user names are included, unless excluded by some other selection criterion.

user-name: Specify the user names of the jobs to select. Because jobs may have identical user names, this value may not identify a specific job. SLTUSRID(user) is equivalent to SLTJOB(*N/user/*N). You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

OMTUSRID

Specifies a list of up to 50 user names to omit. Jobs with any of the user names specified are excluded from the report.

***NONE:** No jobs are excluded based on user name.

user-name: Specify the user names of the jobs to omit. Because jobs may have identical user names, this value may not identify a specific job. OMTUSRID(user) is equivalent to OMTJOB(*N/user/*N). You can specify a generic name for this value. A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

SLTPOOLS

Specifies a list of up to 64 pools to select. Only jobs that run in one of the specified pools are included in the report.

***ALL:** Jobs in all pools are included unless excluded by other selection criteria.

storage-pool-identifier: Specify the number of a pool to select. Valid values range from 1 through 64.

OMTPOOLS

Specifies a list of up to 64 pools to omit. Jobs that run in any of the specified pools are excluded from the report.

Note: The SLTPOOLS and OMTPOOLS parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on their pool.

storage-pool-identifier: Specify the number of the pool to omit. Valid values range from 1 through 64.

SLTSBS

Specifies a list of up to 50 subsystems to select. Only jobs that run in one of the specified subsystems are included in the report.

Note: The SLTSBS and OMTSBS parameters are mutually exclusive.

***ALL:** Jobs in all subsystems are included unless excluded by other selection criteria.

subsystem-name: Specify the name of a subsystem to select.

OMTSBS

Specifies a list of up to 50 subsystems to omit. Jobs that run in any of the specified subsystems are excluded from the report.

Note: The SLTSBS and OMTSBS parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on subsystem.

subsystem-name: Specify the name of a subsystem to omit.

SLTLINE

Specifies a list of up to 50 communications lines to select. Only jobs that use a remote device connected through one of the specified communications lines are included in the report.

Note: The SLTLINE and OMTLINE parameters are mutually exclusive.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

communications-line-name: Specify the name of a communications line to select. This excludes jobs using remote devices connected through other communications lines (or no communications line), even if the controllers to which these devices are attached are specified on the SLTCTL parameter.

OMTLINE

Specifies a list of up to 50 communications lines to omit. Jobs that use a remote device connected through any of the specified communications lines are excluded from the report.

Note: The SLTLINE and OMTLINE parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on communications line.

communications-line-name: Specify the name of a communications line to omit.

SLTCTL

Specifies a list of up to 50 communications controllers to select. Only jobs that use a device connected to one of the specified communications controllers are included in the report.

Note: The SLTCTL and OMTCTL parameters are mutually exclusive.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

controller-name: Specify the name of a communications controller to select.

OMTCTL

Specifies a list of up to 50 communications controllers to omit. Jobs that use a device connected to any of the specified communications controllers are excluded from the report.


Note: The SLTCTL and OMTCTL parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on communications controllers.

controller-name: Specify the name of a communications controller to omit.

SLTFCNARA

Specifies a list of functional areas to select. Only jobs and users identified in one of the functional areas are included in the report.


A functional area is a list of job names and user names previously defined by the user. Refer to the Performance Tools for iSeries  book for more information on functional areas.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

functional-area-name: Specify the name of the functional area to select.

OMTFCNARA

Specifies a list of functional areas to omit. Jobs and users identified in any of the functional areas are excluded from the report.

A functional area is a list of job names and user names previously defined by the user. Refer to the Performance Tools for iSeries  book for more information on functional areas.

***NONE:** Jobs are not excluded based on functional area.

functional-area-name: Specify the name of the functional area to omit.

JOB Specifies the job name used if the job is submitted for batch processing.

Note: If *NONE is specified on the JOBD parameter, this parameter is ignored; job processing is performed interactively.

PRTSYSRPT: The command name is used for the job name.

***MBR:** The name selected for the performance data member on the MBR parameter is used.

job-name: Specify the name used for batch jobs.

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 to be searched.

QPFRJOB: The IBM-supplied Performance Tools job description, QPFRJOB, is used.

job-description-name: Specify the name of an alternative 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 is not available for other use during this time, which could be significant for long jobs.

Examples for PRTSYSRPT

Example 1: Printing a Report

```
PRTSYSRPT MBR(APRIL18)
```

or

```
PRTSYSRPT MBR(APRIL18) SECTION(*ALL)
```

These commands print a complete system report for the performance data member APRIL18 in library QPFRDATA. The report title is the same as the text in the member.

Example 2: Selecting Intervals to Include in Report

```
PRTSYSRPT MBR(NOV1) PERIOD(*SELECT)
  TITLE('Intervals with Highest Response Times')
```

This command prints a system report for the data member NOV1 in library QPFRDATA. The user is presented with the interval-selection screen, which allows sorting of the intervals according to various criteria and the selection of certain intervals to be included in the report. The title of the report is *"Intervals with Highest Response Times."*

Example 3: Selecting Sections to Include in Report

```
PRTSYSRPT MBR(NOV1) SECTION(*DSKUTL)
```

This command prints only the Disk Utilization section of the system report for the data member NOV1.

Error messages for PRTSYSRPT

*ESCAPE Messages

None.

PRTSYSSECA (Print System Security Attributes) Command Description

PRTSYSSECA Command syntax diagram

Purpose

The Print System Security Attributes (PRTSYSSECA) command prints a report of security-related system values and network attributes to a spooled file. The report includes the system value or network attribute name, the current value, and the recommended value.

Restrictions:

You must have *ALLOBJ special authority to use this command.

Error messages for PRTSYSSECA

*ESCAPE Messages

CPFB304

User does not have required special authorities.

PRTTRC (Print Trace) Command Description

PRTTRC Command syntax diagram

Purpose

The Print Trace (PRTTRC) command formats and writes the trace records to >> the selected output file. << The trace records were written to a set of database files by the ENDTRC (End Trace) command >> and PRTTRC is used to format these trace records to a spooled output file or to a database output file. << If the trace records are written to a spooled output file, printer file >> QPSRVTRCJ << is used. The user data for the spooled file will be the same as the value specified for the DTAMBR (Data member) parameter.

Restrictions:

1. To use this command you must have *SERVICE special authority, or be authorized to the Service Trace function of the operating system 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. To use this command you must have authority to the library and the database files within that library where the trace data is stored.
3. If DLTTRC(*YES) is specified, you must have authority to the DLTTRC (Delete Trace Data) command.
4. >> The record format of the database output file must match the record format of the IBM-supplied output file QASCTJFL.
5. <<

Required Parameter

DTAMBR

Specifies the member name for the trace data that you want to print. The member name will be

the same as the trace session identifier specified on the Start Trace (STRTRC) and End Trace (ENDTRC) commands. The member name is the same for each of the physical files that contain the trace data.

member-name: Specify the member name for the trace to print.

Optional Parameters

DTALIB

Specifies the name of the library that contains the set of database files where the collected trace data is stored.

***CURLIB**: The trace data is printed from files in the current library for the job. If no library is specified as the current library for the job, QGPL is used.

library-name: Specify the library that contains the trace data files.

SLTJOB

Specifies which jobs to include in the trace listing. This allows the user to reduce the size of the trace listing by selecting only a subset of the jobs that were part of the trace. Up to ten qualified job names can be specified.

Single Value

***ALL**: All jobs that were part of the trace are included.

Job Name Qualifier

job-name: Specify the name of the job to be included in the trace listing.

generic-job-name*: Specify the generic name of the jobs to be included in the trace listing. 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 job name specifies all jobs with job names that begin with the generic prefix.

Job User Name Qualifier

***ALL**: All jobs that match the specified job name are included.

user-name: Specify the name of the user of the job to be included.

generic-user-name*: Specify the generic user name of the jobs to be included.

Job Number Qualifier

***ALL**: All jobs that match the specified job name and user name are included.

job-number: Specify the job number to further qualify the job name and user name.

DLTTRC

Specifies whether trace data is deleted after it has been printed.

***NO**: The trace data in the database files is saved. The DLTTRC (Delete Trace) command can be used to delete the data when it is no longer needed.

***YES**: The trace data in the database files is deleted after the print has completed.



SORT Specifies how the trace data for each job is sorted in the specified output file.

***THREAD:** The trace data for each job is sorted by thread. If a job has multiple threads, the trace data for each thread is sorted by time.

***TIME:** The trace data for each job is sorted by time. If a job has multiple threads, the trace data for all threads in the job is sorted by time. This can result in the trace output for multiple threads to be intermingled.

OUTPUT

Specifies where the output is sent. For more information on this parameter, see Commonly used parameters.

The possible values are:

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

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

OUTFILE

Specifies the qualified name of the database file to which the trace output is directed. If the output file already exists, the system tries to use it. If the output file does not exist, the system creates a database file with the name specified in the OUTFILE parameter in the specified library. A member is created for the file with the name specified in the OUTMBR parameter. If this command creates the file, the text is "OUTFILE created by PRTRC command", and the public authority is *EXCLUDE.

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

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

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

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

file-name: Specify the name of the file that is used to store the trace information.

OUTMBR

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

Element 1: Member to Receive Output

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

member-name: Specify the name of the output file member use to store the trace information. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

Element 2: Operation to Perform on Member

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

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



Examples for PRITTRC

Example 1: Print and Delete Trace

```
PRITTRC DTAMBR(TRACE8) DTALIB(TRCLIB1)
DLTTRC(*YES)
```

This command formats and prints the trace data contained in database file members named TRACE8 in library TRCLIB1. The trace data members are removed after the trace data spooled file has been written. All jobs which were part of the trace will be part of the trace listing.

Example 2: Print Subset Trace

```
PRITTRC DTAMBR(T123456789) DTALIB(QGPL)
SLTJOB(*ALL/QSYS/QCMN*) DLTTRC(*YES)
```

This command formats and prints the trace data contained in database file members named T123456789 in library QGPL. The trace data members are removed after the trace data spooled file has been written. Only those traced jobs that were started under user profile QSYS and had job names that started with "QCMN" will be part of the trace listing. >>

Example 3: Print Trace and Sort by Time

```
PRITTRC DTAMBR(MYTRACE) DTALIB(MYTRCLIB)
DLTTRC(*YES) SORT(*TIME)
```

This command formats and prints the trace data contained in database file members named MYTRACE in library MYTRCLIB. The trace data members are removed after the trace data spooled file has been written. The trace records are sorted by the time the record was collected. If the traced jobs were multithreaded, the trace output is sorted by job, with all threads in that job sorted by time. The resulting output may have trace information for multiple threads intermingled.

Example 4: Print Trace to an Output File

```
PRITTRC DTAMBR(BIGTRACE) DTALIB(TRACELIB)
DLTTRC(*YES) OUTPUT(*OUTFILE) OUTFILE(MYLIB/MYFILE)
```

This command stores the trace data contained in database file members named BIGTRACE in library TRACELIB into a database output file named MYFILE in library MYLIB. The trace data members named BIGTRACE are removed after the trace data has been written to the database output file. <<

Error messages for PRITTRC

*ESCAPE Messages

CPF39CD

Error occurred during processing of the PRITTRC command.

PRITTRCRPT (Print Trace Report) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

PRITTRCRPT Command syntax diagram

Purpose

The Print Trace Report (PRITTRCRPT) command produces a report that shows resources utilized, exceptions, and state transitions for batch jobs traced through time. This report is based on the trace data collected by the Start Performance Trace (STRPFRTTC) command. This report runs against the specified

member that was created when the Print Transaction Report (PRTTNSRPT) command was run with the *FILE option. This member resides in the QTRJOBT file of the QPFRDATA library.

Required Parameter

MBR Specifies the performance data member used. This name should correspond to the member name specified when the Print Transaction Report (PRTTNSRPT) command was run with the *FILE option.

Optional Parameters

LIB Specifies the library where the performance data is located.

QPFRDATA: The performance data files are located in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library where the performance database files are located.

TITLE Specifies the title for the job trace report that is created.

***MBR:** The member name specified in the MBR parameter is used for the report title.

'report-title': Specifies the title for the job trace report. The title can be up to 50 characters, enclosed in apostrophes.

PERIOD

Specifies the period of time on which to report. The value consists of two lists of two elements each:

```
PERIOD((start-time start-date)
        (end-time end-date))
```

*N may be used to maintain the position in the parameter value sequence in place of an element that precedes the values that are specified. For example, PERIOD(*N (*N 091290)) specifies the ending date and uses the defaults for the other values.

Element 1: Starting Time

One of the following values is used to specify the starting time. Data collected prior to this time on the starting date is not included in the report.

***FIRST:** Data records starting from the beginning of the day (00:00:00) are included.

start-time: Specify the time of the first data record to include in the report. 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

One of the following values is used to specify the starting date. Data collected prior to the starting time on this date is not included in the report.

***FIRST:** Data records starting from the first day of the collection period are included.

start-date: Specify the date of the first data record to include in the report. The date must be entered in the format specified by QDATFMT and, if separators are used, QDATSEP. For instance, the system might have a date format of 'mm/dd/yy'. The month (mm), day (dd), and year (yy) are all required 1- or 2-digit values. The slashes (/) are optional if all six digits are specified. If the slashes are omitted, or if the value is entered from the prompt display, the apostrophes (') are not required.

Element 3: Ending Time

One of the following values is used to specify the ending time. Data collected after this time on the ending date is not included in the report.

***LAST:** Data records through the end of the day (23:59:59) are included in the report.

end-time: Specify the time of the last data record to include in the report. See *start-time* in this parameter for details on how the time must be specified.

Element 4: Ending Date One of the following values is used to specify the ending date. Data collected after the ending time in this date is not included in the report.

***LAST:** Data records through last day of the collection period included in the report.

end-date: Specify the date of the last data record to include in the report. Use the same date format used for the starting date.

JOB Specifies the job name used if the job is submitted for batch processing.

Note: If *NONE is specified on the JOB parameter, this parameter is ignored; job processing is performed interactively.

PRTRCRPT: The command name is used for the job name.

***MBR:** The name selected for the performance data member on the MBR parameter is used.

job-name: Specify the name used for batch jobs.

JOB Specifies the job description used by the jobs that submit the report 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 to be searched.

QPFRJOB: The IBM-supplied Performance Tools job description, QPFRJOB, is used.

job-description-name: Specify the name of an alternative job description.

***NONE:** A batch job is not submitted. Processing continues interactively while the user waits. The user's work station is not available for other use during this time.

Examples for PRTRCRPT

Example 1: Printing a Job Trace Summary Report

```
PRTTTCRPT MBR(JUNE01)
```

This command submits a batch job that generates a Job Trace Summary report using the performance data found in the member JUNE01 of file QTRJOB located in the default library QPFRDATA. The report covers the entire collection period, and the title of the report is set to the name of the database file member.

Example 2: Specifying a Report Time Period

```
PRTTTCRPT MBR(NOV15)
  PERIOD(('0800:00' '11/15/99')
 ('2359:59' '11/15/99'))
  TITLE('Job Trace Reports for November 15')
```

This command submits a batch job that generates a Job Trace Summary report. The performance data comes from member NOV15 of file QTRJOB of the default library QPFRDATA. The report covers the time period 8:00 in the morning to midnight of one day.

Note: The format for the date and time is determined by the system values QDATFMT and, because separators are used in this example, QDATSEP.

Error messages for PRTTTCRPT

*ESCAPE Messages

PFR5515

Cannot access trace data.

PRTTNSRPT (Print Transaction Report) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

PRTTNSRPT Command syntax diagram

Purpose

The Print Transaction Report (PRTTNSRPT) command creates and prints performance reports that show detailed information about the transactions that occurred during the time that the performance data was collected. These reports use trace data collected by using the Start Performance Trace (STRPFRTRC) command. Jobs may be selectively included in the reports or excluded from the reports based on a variety of job details and interval times.

Required Parameter

MBR Specifies the performance data member used. This name should correspond to the member name specified on the MBR parameter of the End Performance Trace (ENDPFRTRC) command.

Optional Parameters

LIB Specifies the library where the performance data is located.

QPFRDATA: The performance data files are located in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library where the database files are located.

TITLE Specifies a title that prints in the heading at the top of each page of the report.

***BLANK:** No title is printed on the transaction report.

'report-title': Specify a title of up to 50 characters enclosed in apostrophes.

RPTTYPE

Specifies the type of transaction analysis report that is printed. A list of report types can be requested so that both summary level and transaction detail reports can be requested at the same time. The *TNSACT and *TRSIT reports are quite detailed. Use of these values should be combined with selection of specific jobs, users and/or time intervals.

***SUMMARY:** A summary level report is printed.

***TNSACT:** The transaction detail report is printed.

***TRSIT:** The transition detail report is printed.

***FILE:** A transaction summary, job summary, and job trace database file members are created. The summaries exist in the QTRTSUM, QTRJSUM, and QTRJOBT in the library specified for the LIB parameter. The member names are the names specified on the MBR parameter. The data in any existing member is replaced as a result of this command. This option is used to build field level database files that are processed by user-defined programs and the Print Job Trace Report (PRTTTRCRPT) command.

***TRCDTA:** A database file version of the trace data file QAPMDMPT is created. The database file is named QTRDMPT and is a field-level database file which can be processed by user-defined programs.

PERIOD

Specifies the times when transactions are reported. If PERIOD is not specified, the following values are assumed:

```
PERIOD((*FIRST) (*LAST))
```

Element 1: Starting Time

One of the following is used to specify the starting time after which the data must be measured to be included in the report. Data measured before the specified time is not shown.

***FIRST:** Transactions are reported beginning with the first one recorded.

start-time: Specify the time after which measured data is included in the report. 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: Ending Time

One of the following is used to specify the ending time before which the data must be measured to be included in the report.

***LAST:** Transactions are reported ending with the last one recorded.

end-time: Specify the time after which measured data is not included in the report. See *start-time* in this parameter for details on how the time must be specified.

OPTION

Specifies additional options that can be applied to the transaction report.

***SS**: An additional set of system summary reports are included. These reports are included if `RPTTYPE(*SUMMARY)` is specified.

***SI**: Only interactive jobs are selected.

***OZ**: All jobs that have zero transactions are omitted from the `*SUMMARY` report.

***EV**: Event wait is considered as a transaction boundary. This is useful in the analysis of communications jobs.

***HV**: SLIC tasks are listed on the `*SUMMARY` report.

***DI**: The trace records for the display I/O transaction boundary are counted as transactions, instead of wait-to-active state transitions.

***DQ**: The trace records for the data queue transaction boundary are counted as transactions, instead of wait-to-active state transitions.

If you specify `*DI` and `*DQ`, the Transaction Report uses a combination of display I/O and data queue transaction boundary records instead of wait-to-active state transitions to tabulate transactions.

If you do not specify either `*DI` or `*DQ`, the Transaction Report uses the traditional wait-to-active state transitions to tabulate transactions.

DETAIL

Specifies whether you want the report to provide detailed job information at the job level or the thread level.

***JOB** Specifies that you want detailed information at the job level.

***THREAD** Specifies that you want detailed information at the thread level.

SLTJOB

Specifies which jobs are included in the report. This allows you to narrow the scope of the report to certain jobs through job selection. Specifies either the special value `*ALL` or a list of qualified names of jobs to select.

Element 1: Job name

***ALL**: All the jobs are included in the transaction report.

job-name: Specify one or more job names to be included in the transaction report. A generic job name may be specified in the form `NAME*`.

Note: The job name is not a fully qualified job name. It is the ten-character job name portion of the qualified name. The job number is allowed on this parameter. The use of job name and job number cannot be mixed; one or the other must be used on a given request.

Element 2: Thread

***ALL**: All threads are included, unless excluded by some other selection criterion.

thread-identifier: Specify the thread identifier to select. Because some jobs can have identical thread identifiers, this value may not identify a specific thread.

OMTJOB

Specifies which jobs are omitted from the report. This allows the user to narrow the scope of the transaction report to certain jobs through job omission.

The SLTJOB and OMTJOB parameters are mutually exclusive, so the default must be used for at least one of them.

Element 1: Job name

***NONE:** None of the jobs are excluded from the transaction report.

job-name: Specify one or more job names to be excluded from the transaction report. A generic job name may be specified in the form NAME*.

Note: The job name is not a fully qualified job name. It is the ten-character job name portion of the qualified name. The job number is allowed on this parameter. The job name and job number cannot be mixed, one or the other must be used on a given request.

Element 2: Thread

***ALL:** All threads are included, unless excluded by some other selection criterion.

thread-identifier: Specify the thread identifier to select. Because some jobs can have identical thread identifiers, this value may not identify a specific thread.

SLTUSRID

Specifies a list of up to 50 user names to select. Only jobs with one of the specified user names are included in the report.

***ALL:** Jobs with all user names are included, unless excluded by some other selection criterion.

user-name: Specify the user names of the jobs to select. Because jobs may have identical user names, this value may not identify a specific job. You can specify a generic name for this value. SLTUSRID(user) is equivalent to SLTJOB(*N/user/*N). A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

OMTUSRID

Specifies a list of up to 50 user names to omit. Jobs with any of the user names specified are excluded from the report.

***NONE:** Jobs are not excluded based on user name.

user-name: Specify the specific or generic user name of the job to omit. Because jobs may have identical user names, this value may not identify a specific job. You can specify a generic name for this value. SLTUSRID(user) is equivalent to SLTJOB(*N/user/*N). A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid 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 name, the system assumes it to be the complete object name.

SLTPOOLS

Specifies a list of up to 64 pools to select. Only jobs that run in one of the specified pools are included in the report.

***ALL:** Jobs running in all pools are included, unless excluded by other selection criteria.

storage-pool-identifier: Specify the number of a pool to select. Valid values range from 1 through 64.

OMTPOOLS

Specifies a list of up to 64 pools to omit. Jobs that run in any of the specified pools are excluded from the report.


Note: The SLTPOOLS and OMTPOOLS parameters are mutually exclusive.

***NONE:** Jobs are not excluded based on their pool.

storage-pool-identifier: Specify the number of the pool to omit. Valid values range from 1 through 64.

SLTFCNARA

Specifies a list of functional areas to select. Only jobs and users identified in one of the functional areas are included in the report.


A functional area is a list of job names and user names previously defined by the user. Refer to the Performance Tools for iSeries  book for more information on functional areas.

***ALL:** All jobs are included, unless excluded by some other selection criterion.

functional-area-name: Specify the name of the functional area to select.

OMTFCNARA

specifies a list of functional areas to omit. Jobs and users identified in any of the functional areas are excluded from the report.

A functional area is a list of job names and user names previously defined by the user. Refer to the Performance Tools for iSeries  book for more information on functional areas.

***NONE:** Jobs are not excluded based on functional area.

functional-area-name: Specify the name of the functional area to omit.

JOB Specifies the job name used if the job is submitted for batch processing.

Note: If *NONE is specified on the JOB parameter, this parameter is ignored; job processing is performed interactively.

PRTTNSRPT: The command name is used for the job name.

***MBR:** The name selected for the performance data member on the MBR parameter is used.

job-name: Specify the name used for batch jobs.

JOB 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 to be searched.

QPFRJOB: The IBM-supplied Performance Tools job description, QPFRJOB, is used.

job-description-name: Specify the name of an alternative 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 is not available for other use during this time, which could be significant for long jobs.

Examples for PRTTNSRPT

Example 1: Printing a Summary Transaction Report

```
PRTTNSRPT MBR(TUESAM)
```

This command produces a summary transaction report. The data input to the report is all the data that exists in member TUESAM in library QPFRDATA. The request is sent to batch. The report output is directed to the output queue specified in the job description, QPFRJOB.

Example 2: Printing a Transaction Detail Report

```
PRTTNSRPT MBR(TUESAM) RPTTYPE(*TNSACT)
          SLTJOB(WS01)
```

This command produces a transaction detail report for the selected job, WS01. The request is sent to batch. The report output is directed to the output queue specified in the job description, QPFRJOB.

Error messages for PRTTNSRPT

*ESCAPE Messages

None.

PRTRGPGM (Print Trigger Program) Command Description

PRTRGPGM Command syntax diagram

Purpose

The Print Trigger Program (PRTRGPGM) command lists the programs which have been defined as a trigger program for the physical files in the specified library.

This command prints two reports for a library. The first report (Full Report) contains all of the trigger programs associated with files in the specified library. The second report (Changed Report) contains the trigger programs that now appear in the specified library and were not in the library when the PRTRGPGM command was previously run for the library. If the PRTRGPGM command was not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional trigger programs are in the specified library, the 'Changed Report' is printed but there are no objects listed. Changing the trigger time, trigger event or trigger update condition for a trigger program will not cause a 'Changed Report' to be generated.

A program is not listed in the 'Changed Report' when the program itself is changed. Therefore, you should periodically review the entire list of objects that adopt to understand their current function.

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

Required Parameter

LIB Specifies the name of the library to search for files that have trigger programs.

***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:** All user libraries are searched.

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

Optional Parameter

CHGRPTONLY

Specifies whether just the changed report should be printed.

***NO:** The full and changed reports are printed.

***YES:** Only the changed report is printed.

Example for PRTRGPGM

```
PRTRGPGM LIB(*ALL)
```

This command searches all files in all libraries and produces the full and changed trigger program reports.

Error messages for PRTRGPGM

*ESCAPE Messages

CPFB304

User does not have required special authorities.

PRTUSROBJ (Print User Objects) Command Description

PRTUSROBJ Command syntax diagram

Purpose

The Print User Objects (PRTUSROBJ) command allows you to print a report of the objects in a library that are not created by IBM. Objects are included in the report if the "Created by user" attribute is not *IBM or QLPINSTALL. Use this command to check for user created objects that are in libraries intended for use only by IBM. For example, you may want to run this program for library QSYS to determine if it contains any non-IBM (user) objects.

This command prints two reports for a library. The first report (Full Report) contains all of the objects that have not been created by IBM. The second report (Changed Report) contains the objects that now appear

in the specified library and were not in the library when the PRTUSROBJ command was previously run for the library. If the PRTUSROBJ command was not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional objects have been added to the library that were not created by IBM, then the 'Changed Report' is printed but there are no objects listed.

Note: Some objects created by IBM will still appear in this report. For example, objects created by a PTF exit program are included. Objects are excluded from the report only when their 'Created by user' attribute is either *IBM or QLPINSTALL.

Restrictions: You must have *ALLOBJ special authority to use this command.

Required Parameter

LIB Specifies the name of the library to search for objects that were not created by IBM.
library-name: Specify the name of the library to be searched.

Optional Parameter

CHGRPTONLY

Specifies whether just the changed report should be printed.

***NO:** The full and changed reports will be printed.

***YES:** Only the changed report will be printed.

Example for PRTUSROBJ

```
PRTUSROBJ LIB(QSYS) CHGONLY(*NO)
```

The library QSYS will be searched for any objects that were not created by IBM and the full and changed reports will be created.

Error messages for PRTUSROBJ

*ESCAPE Messages

CPFB304

User does not have required special authorities.

PRTUSRPRF (Print User Profile) Command Description

PRTUSRPRF Command syntax diagram

Purpose

The Print User Profile (PRTUSRPRF) command allows you to print a report containing information for the user profiles on the system. Four different reports can be printed. One contains authority type information, one contains environment type information, one contains password type information, and one contains password level type information.

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

Optional Parameters

TYPE Specifies the type of information that can be printed for the selected user profiles.

***ALL:** All of the reports are printed for the selected user profiles.

***AUTINFO:** A report containing the authority type information for the selected user profiles is printed.

***ENVINFO:** A report containing the environment type information for the selected user profiles is printed.

***PWDINFO:** A report containing the password type information for the selected user profiles is printed.

***PWDLVL:** A report containing the password level type information for the selected user profiles is printed. This report can be used to determine which user profiles have passwords that are used at the different password levels.

SELECT

Specifies what criteria is used to select the user profiles to include in the report.

***SPCAUT:** User profiles are selected for the report based on special authorities.

***USRCLS:** User profiles are selected for the report based on user class.

***MISMATCH:** User profiles are selected for the report based on their special authorities not being the default values assigned to their user class.

Note:

The defaulted special authorities for user classes were changed in Version 3, Release 7, Modification 0. Therefore, when running this report for profiles created prior to V3R7M0, you may notice a larger than expected number of profiles that do not match the default values.

SPCAUT

If *SPCAUT was specified for the SELECT parameter, it specifies which special authorities should be used to select users. User profiles with any of the special authorities specified for this parameter are included in the report. A maximum of 9 special authorities can be specified.

***ALL:** All user profiles are included in the report.

***ALLOBJ:** User profiles with *ALLOBJ special authority are included in the report.

***AUDIT:** User profiles with *AUDIT special authority are included in the report.

***JOBCTL:** User profiles with *JOBCTL special authority are included in the report.

***IOSYSCFG:** User profiles with *IOSYSCFG special authority are included in the report.

***SAVSYS:** User profiles with *SAVSYS special authority are included in the report.

***SECADM:** User profiles with *SECADM special authority are included in the report.

***SERVICE:** User profiles with *SERVICE special authority are included in the report.

***SPLCTL:** User profiles with *SPLCTL special authority are included in the report.

***NONE:** User profiles with no special authorities are included in the report.

USRCLS

If *USRCLS was specified for the SELECT parameter, it specifies that user classes should be used to select users. User profiles with a user class that is specified for this parameter are included in the report. A maximum of 5 user classes can be specified.

***ALL:** All user profiles are included in the report.

***USER:** User profiles with *USER user class are included in the report.

***SYSOPR:** User profiles with *SYSOPR user class are included in the report.

***PGMR:** User profiles with *PGMR user class are included in the report.

***SECADM:** User profiles with *SECADM user class are included in the report.

***SECOFR:** User profiles with *SECOFR user class are included in the report.

Example for PRTUSRPRF

```
PRTUSRPRF TYPE(*ALL) SELECT(*SPCAUT)
          SPCAUT(*ALLOBJ *SECADM)
```

This command creates all four reports for user profiles that have either *ALLOBJ or *SECADM special authority.

Error messages for PRTUSRPRF

*ESCAPE Messages

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

PGM (Program) Command Description

PGM Command syntax diagram

Purpose

The Program (PGM) command is used in a CL program source file to identify the start of a CL program that is to be compiled and to specify the parameters that are to be received by the program after it is compiled. If a PGM command is used, it must be the first command in the program source file; if a PGM command is not used, a PGM command without parameters is assumed. The name of the program is specified in the Create Control Language Program (CRTCLPGM) command that is used to create the CL program.

The PGM command also specifies any parameters that are passed to the program when it is called for processing by another program. For information about how constants are passed, see the PARM parameter description under the CALL command.

This program can be called by a Call (CALL) or Transfer Control (TFRCTL) command, or by a routing entry in a subsystem description. When the program is called by a CALL or TFRCTL command, the specified parameters can be passed to it.

Parameters defined in this command must be passed when the program is called. The parameters passed must be of the type, length, and order specified in this command. Each of the parameter values passed to this program can be a character string, a numeric value, or a CL variable. When received by this program, each value is given a different CL variable name. Each CL variable name must be defined in the CL source file by a separate DCL (Declare) command before the program is compiled. Up to 40 parameters can be passed.

ILE programs and procedures will not detect parameter mismatches between the calling programs or procedure and the called program or procedure. If the calling procedure passes more parameters than the called procedure expects, the called procedure will ignore the extra parameters. If the calling procedure passed fewer parameters than are specified on the called procedure PGM command, the result may be unexpected.

Restriction: This command is valid only in a CL program.

Optional Parameter

PARM Specifies the names of one or more CL variables that are to receive the parameter values passed to this program. Specify a CL variable name for each of the values to be received from the calling program; the name must start with an ampersand (&). Null values, *N, cannot be specified for any parameter. The parameter values are associated with the variables in the PARM parameter in the order in which they were specified on the CALL or TFRCTL commands. The type and length of each value passed must have matching attributes in the calling and receiving programs. However, for character constants, the receiving program can specify a shorter length; when this is done, the character string passed is truncated to the length declared in the receiving program. For information on how each data type is passed, see the description of the PARM parameter in the CALL command.

Note:

If a parameter value is to be changed by a CL program or specified as a variable on a CL command, it must be in writeable storage. For example, in C/400, strings may be read only. If a read only string is passed as a parameter to a CL program, and the CL program attempts to change the value of the variable or uses the variable on a CL command, the CL program will fail.

Examples for PGM

Example 1: Program Containing No Parameters

```
PGM
*
*
*
ENDPGM
```

This PGM command is the first command in a CL program source file for a program that contains no parameters.

Example 2: Program Containing Two Parameters

```
PGM PARM(&X &Y)
```

This is the first command in a program source file for a program that contains two parameters, &X and &Y, that have values passed to them from the calling program.

Example 3: Program Containing Two Parameters in Positional Form

```
PGM (&PARAM1 &PARAM2)
```

This is the first command in a program source file for a program that specifies two parameters in positional form, &PARAM1 and &PARAM2. When this program is called, the calling program passes, through the CALL or TFRCTL command, the values to be assumed for &PARAM1 and &PARAM2.

Error messages for PGM

QSH (Start QSH) Command

QSH syntax diagram

For the description of the QSH command, see the STRQSH (Start QSH) command description.

QRYDST (Query Distribution) Command Description

QRYDST Command syntax diagram

Purpose

The Query Distribution (QRYDST) command allows a request for distribution information for the user or on behalf of another user.

Restrictions:

1. If the current user of this command requests distribution information for another user, the current user must have been granted the authority to work on behalf of the other user by means of the Grant User Permission (GRTUSRPMN) command.
2. If USRID(*ALLAUT) is specified and the current user of this command does not have the authority to work on behalf of the other user, only the information about the current user's distributions is returned.
3. DLTSTS does not apply to incoming distributions. When OPTION(*IN) is specified, the DLTSTS parameter is ignored.
4. The requester of the command (the user who is signed on) must be enrolled in the system distribution directory.
5. Personal distribution cannot be questioned if the requester is working on behalf of another user.

Note:

The formats of the output files must be OSLIN or OSLOUT. These formats are defined in the physical files QAOSILIN or QAOSILOT, respectively. These files are located in library QSYS. Users can specify the Create Duplicate Object (CRTDUPOBJ) command to create duplicates of these files for their libraries. If the user's library does not contain the files, the files are created when the command is run.

Optional Parameters

USRID

Specifies the user ID and address of the user making this request. The user specifying this command must have authority to work on behalf of the user specified in this parameter if the named user ID and address differs from that of the current user.

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

***ALLAUT:** Distribution information is returned for users who have given the current user of this command the authority to work on their behalf.

Element 1: User ID

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

Element 2: User Address

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

OPTION

Specifies the type of distribution information that is returned.

***IN:** Information about incoming distributions is returned. If an output file is specified in the `OUTFILE` parameter, one incoming distribution information record per distribution is written to the output file.

***OUT:** Information about outgoing distributions is returned. If an output file is specified, N outgoing distribution information records per distribution are written to the output file. N is the number of original receivers of the distribution or the number of receivers that have distribution errors.

DLTSTS

Specifies whether the status being kept for outgoing distributions is deleted from the system. This can be error or confirmation of delivery status.

***NO:** The distribution status is not deleted from the system. The information is kept by the system and can be returned by a request using the QRYDST command.

***YES:** The distribution status is deleted if all receivers are at returned status or completed confirmation of delivery status.

Note:

Other products use this status information. Care should be taken not to delete information used by other products to track their distributions.

Status is deleted by the system if all receivers have returned status or the status is returned to another product (such as the OfficeVision) for this user.

OUTFILE

Specifies the qualified name of the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library.

For incoming distributions, the system uses QAOSILIN in QSYS with the format name of OSLIN as a model.

For outgoing distributions, the system uses QAOSILOT in QSYS with the format name of OSLOUT as a model.

The authority for users with no specific authority to the output file is *EXCLUDE. More information on defining the format of database files (output file) is in the Office Services Concepts and Programmer's Guide book.

***NONE:** The output is not directed to a database file. If *NONE is specified, the output from this command is a completion message containing the number of distributions on the DIA Distribution Recipient Index (*DRX) for the specified category and user.

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

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

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

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

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

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

STATUS

Specifies the mail entry status of the distribution for which the user is requesting information. This parameter is valid only if *IN is specified on the OPTION parameter and an output file is specified on the OUTFILE parameter.

***ALL:** Distribution information is returned regardless of the distribution's mail entry status.

***NEW:** Distribution information is returned only for distributions with a mail entry status of NEW.

***OLD:** Distribution information is returned only for distributions with a mail entry status of OLD. A mail entry status of OLD indicates that the distribution has been queried once but has not been processed.

***OPENED:** Distribution information is returned only for distributions with a mail entry status of OPENED.

***UNOPENED:** Distribution information is returned for distributions with a mail entry status of OLD or NEW.

***LOCAL:** Distribution information is returned only for distributions with a mail entry status of LOCAL. A mail entry status of LOCAL indicates that the distribution has been filed on the local system.

***REMOTE:** Distribution information is returned only for distributions with a mail entry status of REMOTE. A mail entry status of REMOTE indicates that the distribution has been filed on a remote system.

***FILEPND:** Distribution information is returned only for distributions with a mail entry status of FILEPND. A mail entry status of FILEPND indicates that the distribution is being filed on a local or remote system but the filing has not been completed.

***DELETED:** Distribution information is returned only for distributions with a mail entry status of DELETED. A mail entry status of DELETED indicates that the document referred to by the distribution has been deleted.

***DAMAGED:** Distribution information is returned only for distributions with a mail entry status of DAMAGED. A mail entry status of DAMAGED indicates that the document referred to by the distribution is damaged.


CMDCHRID

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

Application Display Programming  book.

Note:

This value translates the USRID parameter to the character set and code page set 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.

Example for QRYDST

```
QRYDST USER(*CURRENT) OPTION(*IN)
      OUTFILE(*CURLIB/MYFILE) OUTMBR(*FIRST *ADD)
```

This command requests information about incoming distributions for the current user of this command. The output is directed to the database file MYFILE in the user's current library and is added to the first member in that file.

Error messages for QRYDST***ESCAPE Messages****CPF900B**

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

CPF900C

Sign on and verify of user failed.

CPF905C

Error occurred trying to find a translation table.

CPF907C

&1 requested distributions completed, acknowledge failed.

CPF9096

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

CPF9097

Query distribution request failed.

CPF9845

Error occurred while opening file &1.

CPF9847

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

QRYDOCLIB (Query Document Library) Command Description

QRYDOCLIB Command syntax diagram

Purpose

The Query Document Library (QRYDOCLIB) command allows the user to search for documents within the document library to which the user is authorized and to copy information about the documents that satisfy the search request into a database file for processing.

When the QRYDOCLIB command is run, a document list object is created. The document list object is created regardless of whether an output file is produced unless the user specifies *NONE for the DOCL parameter. This document list object is used by the OfficeVision product as well as the SAVDLO command.

Restriction: The current user of this command must have the authority to work on behalf of the specified user ID address. To work on behalf of other users, the user must have special permission granted with the Grant User Permission (GRTUSRPMN) command. Several QRYDOCLIB commands can run concurrently. If the document list name or the output file is the same on more than one QRYDOCLIB command, errors may occur.

Note:

The format of the output file must be the same as OSIQDL of the system file, QSYS/QAOSIQDL.

Optional Parameters

USRID

Specifies the user ID and address of the user for whom this request is made.

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

Element 1: User ID

user-ID: Specify the user ID of the user for whom the documents are requested.

Element 2: User Address

user-address: Specify the user address of the user for whom the documents are requested.

OUTFILE

Specifies the name of the database file to which the output is directed. If the output file does not exist, this command creates a database file in the specified library. If the file is created by this function, the descriptive text is *OUTFILE created by QRYDOCLIB* and the authority for users without specific authority to the file is *EXCLUDE.

***NONE:** The output is not directed to a database file. A message is returned to the user indicating the number of documents that satisfied the search request. More information on defining the format of database files (output files) is in the Office Services Concepts and Programmer's Guide.

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

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

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

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

database-file-name: Specify the qualified name of the database file that receives the output. This file can be reused when other QRYDOCLIB commands are entered. Output from the file can start at the start of the file member or be added to the file, as specified in the OUTMBR parameter. The IBM-supplied database file, QSYS/QAOSIQDL, cannot be specified.

OUTMBR

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

Element 1: Member to Receive Output

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

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

Element 2: Operation to Perform on Member

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

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

OUTDTATYP

Specifies the type of information about the selected documents that is written to the output file if one is specified on the OUTFILE parameter.

***DFT**: A system-created name is used as the default name. The document information record is written to the output file. Specifying OUTDTATYP(*DFT) is equivalent to specifying OUTDTATYP(*DOCD). The following record is written to the output file:

Record Code	Description
105	Document Description

***ALL**: All record formats about the document are written to the output file. These record formats include the following:

Record Code	Description
105	Document Description
110	Creation Date
115	Expiration date
120	Document date
125	File date
130	Last document change date
135	Action due date
140	Completion date
145	Author
150	Copy list
155	Document class
160	File cabinet reference
165	Subject
170	Keyword
175	Reference

180	Status
185	Project
190	Last indexed date
195	Last content revision date
200	Last used date
500	Interchange document profile data

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

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

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

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

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

***CHGDATE:** The date of the last change to the document is written to the output file.

***USEDATE:** The date of the last usage of the document is written to the output file.

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

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

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

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

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

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

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

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

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

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

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

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

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

***IDP:** The interchange document profile is written to the output file.

DOCL Specifies the name of the document list. A document list is an object that contains a pointer to each document in the document library that the current user is authorized to request. This list is a copy of the library at the time the search was run. As documents are deleted from or added to the library, the document list is not updated. The document library list name is specified with the name of the user requesting the search. Users can use identical document names. For example, Tom could name his list SALES and so could Mary. The system knows the lists as TOM_SALES and MARY_SALES.

***DFT:** A system-created name is used as the default document list name. The default list is the same as the user ID entered in the USRID parameter (TOM_TOM or MARY_MARY).

***NONE:** No document list is created.

document-list-name: Specify the name of the document list. Up to 8 characters can be used.

TEXT Specifies the text that briefly describes the document list 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.

TIMLMT

Specifies the maximum run time (in minutes) the search request can use.

***NOMAX:** No time limit for the search is set. All qualified documents are searched.

time-limit: Specify the maximum time limit (in minutes) that the search runs. Up to 4 digits, ranging from 1 through 9999, can be specified for the number of minutes. A two-hour limit is specified as TIMLMT(120). If the search has not been completed when the time limit is reached, the search ends with an informational message followed by a completion message. The output file, and if specified the document list, will contain the documents found within the specified time limit.

SELLMT

Specifies the number of documents to select in the search.

***NOMAX:** No document limit for the search is set. All qualified documents are selected.

selection-limit: Specify the maximum number of documents to select. A value ranging from 1 through 32767 can be specified. If there are more query requests than the set limit, the document list and the output file contain the information about the selected documents up to this limit. If there is at least one more document that meets the query definition, an informational message is sent before the completion message.

FLR Specifies the folders that are searched to locate the documents that match the query definition specified in the QRYDFN parameter.

***ALL:** All of the folders on the system are searched to locate the documents.

***NONE:** Documents not located in any folder are searched.

folder-name: Specify the name of the folders to search to locate the documents. A folder name can consist of a series of folder names (FLR1/FLR2/and so on) if the documents being searched for are located in a folder contained in another folder. A maximum of 100 folders can be specified and each folder name can be a maximum of 63 characters in length.

SCHSUBFLR

Specifies whether subfolders of the folder specified on the FLR parameter are searched.

***NO:** Subfolders are not searched.

***YES:** Subfolders of the specified folder are searched.

QRYDFN

Specifies whether a query definition is used to select documents. The values specified on this parameter are used to search the document library. If values other than *NONE are specified on both the QRYDFN parameter and QRYTXT parameter, only documents that match both sets of values are selected.

***NONE:** No query definition is used to select the documents. All documents that are owned or accessible are selected.

***IF:** A query definition is used to select the documents.

To specify the conditions under which documents are selected, a set of values is specified for each condition. Each set contains four values.

1. The name of the document profile parameter to be compared
2. One of the relational operator values
3. The comparison value
4. One of the logical operators, *AND or *OR

Values 1 and 3 are compared for the relationship specified by value 2.

Each QRYDFN relational set must be enclosed in parentheses. A maximum of 49 sets (conditions) can be specified.

Element 1: Profile Parameters

profile-parameter: Specify a value representing the profile parameters to be compared.

Value	Description
*DOCD	Document description
*DOCCLS	Document class
*SUBJECT	Document subject
*DOCDATE	Document date
*EXPDATE	Expiration date
*FILDATE	File date
*CRTDATE	Create date
*ACTDATE	Action due date
*CMPDATE	Completion date
*CHGDATE	Last document change date
*USEDATE	Last used date
*DOCTYPE	Document type
*IDXDATE	Last indexed date
*REVDATE	Last content revision date
*AUTHOR	Document author
*KWD	Keyword
*CPYLST	Copy list
*ALWRPL	Allow document replacement
*OWNER	Document owner
*PROJECT	Document project
*REF	Reference
*STATUS	Document status
*ASP	Auxiliary storage pool ID

Element 2: Relational Operator

relational-operator: Specify the operator that indicates the relationship that must exist between the profile parameter contents in the document and the value specified as the third part of this QRYDFN relation for the relation to be true. The operators that can be used are:

Value	Description
*EQ	Equal
*GT	Greater than
*LT	Less than
*NE	Not equal
*GE	Greater than or equal
*NL	Not less than
*LE	Less than or equal

*NG	Not greater than
*CT	Contains
*BG	Begins

The *CT operator performs a context search. It asks the system to determine whether the character string specified by this value is contained anywhere in the profile. For example, a query request of (*IF ((*SUBJECT *CT 'Sales')))) would find a match for subjects that were: '1985 Car Sales', 'Sales Awards', 'Salesperson Training Courses', 'Book of Sales Do's and Don'ts'.

The *BG operator performs a search that compares the specified value with the start of the profile parameter. The profile parameter is truncated or extended as necessary to match the length of the specified value. It asks the system to determine whether the character string specified by the value is contained at the start of the profile parameter. For example, a query request of (*IF ((*SUBJECT *BG 'Sales')))) would find a match for subjects of: 'Sales Awards', 'Salesperson Training Courses', and 'Sales by Phone'.

Some operators are excluded from being used with some profile parameters. If the excluded operators are specified in restricted profile parameters, the request is rejected with a diagnostic message followed by an escape message. Two cases are invalid:

- The *ALWRPL (allow document replacement) is a YES/NO value. The *EQ operator is the only operator allowed to have *ALWRPL.
- The *CT and *BG operators are not allowed with the *ASP value or date values such as *CRTDATE and *EXPDATE.

Element 3: Compare Values

compare-value: Specify the value to compare with the contents of the specified profile parameter. The parameter value is specified in apostrophes if it contains blanks or special characters.

The *ALWRPL field has two special values: *YES and *NO. When these are specified with the *ALWRPL field, they are changed to internal values for the indicator. When specified for the text field, *YES and *NO retain their original values.

The *OWNER field is an 8-character user ID followed by its address. Trailing blanks cannot be omitted from the user ID. For example, if the user ID is JMDOE and the address is SYSTEM1, the query request would be (*IF ((*OWNER *EQ 'JMDOE SYSTEM1'))). If the user ID is JIMSMITH, the query request would be (IF ((*OWNER *EQ 'JIMSMITHSYSTEM1'))).

Dates must be specified in the system date format.

The allowable length for the search fields is limited by the Document Interchange Architecture (DIA) search database. When the length of the value is greater than the maximum, the value is truncated to the allowed length. The maximum lengths are:

Value	Maximum Length
*DOCD	44 characters
*DOCCLS	16 characters
*SUBJECT	60 characters
*AUTHOR	20 characters
*KWD	60 characters
*CPYLST	60 characters
*OWNER	16 characters
*REF	60 characters
*STATUS	20 characters
*PROJECT	10 characters

For all operators except *CT and *BG, if a value that is shorter than the profile parameter value is specified, it is padded on the right with blanks to match the length of the profile parameter.

The case (upper, lower, or mixed) of the letter characters used to enter the original parameter value or the case of the comparison value does not matter. The system changes both the specified comparison value and the original parameter value to upper case before making a comparison. If the original document had been filed with a subject equal to 'Salesperson Training Courses', any of the following would be a match:

```
(*IF ((SUBJECT *EQ 'salesperson training
      courses')))
(*IF ((SUBJECT *EQ 'Salesperson Training
      Courses')))
(*IF ((SUBJECT *EQ 'SALESPERSON TRAINING
      COURSES')))
(*IF ((SUBJECT *CT 'PERSON')))
(*IF ((SUBJECT *CT 'person')))
(*IF ((SUBJECT *BG 'SALES')))
(*IF ((SUBJECT *BG 'sales')))
```

Element 4: Logical Operator

***AND:** The profile parameter value relational groups on both sides of the *AND value must be satisfied before a document is selected.

***OR:** If the parameter value relational group on either side of the *OR value is satisfied, the document is selected.

The logical operators are used to group conditions. The first AND operator encountered signifies that a condition group starts with the condition immediately preceding the AND operator. Subsequent conditions with the AND operator are added to the condition group. The first condition encountered that contains the OR operator (when no more matrix entries exist) ends the condition group.

For example:

```
QRYDFN(*IF
  ((COND1 *OR)
   (COND2 *AND) <---- |
   (COND3 *AND) <---- | --Group 1
   (COND4 *AND) <---- |
   (COND5 *OR)
   (COND6 *OR)
   (COND7 *AND) <---- |
   (COND8 *AND) <---- | --Group 2
   (COND9))) <---- |
```

The previous example could be expressed:

```
(cond1) | (cond2 & cond3 & cond4 & cond5)
| cond6 | (cond7 & cond8 & cond9)
```

Because the AND operator is used to group conditions, the following logical expression cannot be used by the QRYDFN parameter.

```
(cond1 | cond2) & (cond3 | cond4)
```

Examples for QRYDFN

```
QRYDFN(*IF ((*ACTDATE *GE '6/1/1995' *AND)
  (*AUTHOR *EQ 'JOHN HARKER' *OR)
  (*ACTDATE *GE '6/1/1995' *AND)
  (*PROJECT *EQ 'MGMT')))
```

All documents that have an action date later than or equal to 6/1/1995 with author John Harker or project MGMT are selected. This request is made up of two condition groups (AND sets). The first group requires that the documents selected 1) must have an action date of 6/1/1995 or later and

2) must have John Harker as the author. The second group requires that the documents selected 1) must have an action date of 6/1/1995 or later and 2) must be part of project MGMT. If either of these condition groups is true, the document is selected.

```
QRYDFN (*IF ((*AUTHOR *EQ 'SUSAN MICKLE' *OR)
(*PROJECT *EQ 'BASEBALL' *AND)
(*CMPDATE *GE '6/1/89'))))
```

All documents that with the author of Susan Mickle or that have a project of BASEBALL with a completion date on or after 6/1/89 are selected.

```
FLR('RECORDS/SPORTS/BASEBALL/NATIONAL')
QRYDFN(*IF ((*DOCD *CT 'giants' *AND)
(*FILDATE *GE '1/1/1984' *AND)
(*FILDATE *LE '10/1/1986'))))
```

If the word 'giants' is contained somewhere in the document description profile parameter, and if the document file date is between 1/1/1984 and 10/1/1986, the document is selected.

The NATIONAL folder is contained in the BASEBALL folder, which is in the SPORTS folder, which is contained in the RECORDS folder.

```
QRYDFN(*IF ((*EXPDATE *LE '1/1/86'))))
```

All documents accessible by the user doing the search where the expiration date is less than or equal to 1/1/86, are selected.

```
QRYDFN(*IF ((*AUTHOR *EQ 'ANDERSON' *AND)
(*DOCCLS *EQ 'account 431-2' *AND)
(EXPDATE *LE '1/1/86'))))
```

All documents accessible by the user doing the search, when all three conditions on the author, document class, and expiration date profile parameters are met, are selected.

```
QRYDFN(*IF ((*KWD *EQ 'alphabet soup' *OR)
(*KWD *CT Campbells *OR)
(*KWD *BG 'good for you'))))
```

All documents accessible by the user doing the search, when any one of the three keyword tests is satisfied, are selected.

QRYTXT

Specifies the text search values used to select documents. The values specified on this parameter are used to search the text search index. If values other than *NONE are specified on both the QRYDFN parameter and the QRYTXT parameter, only documents that match both sets of values are selected.

***NONE:** No text search values are used to select the documents.

***IF:** Text search values are used in the document search.

Note: When *IF is specified, ordering is not allowed. *NONE must be specified on the ORDER parameter.

To specify the conditions under which documents are selected, a set of values is specified for each condition. Each set contains four values:

1. A phrase, which the system compares to entries in the text search index
2. One of the type of phrase matching values
3. One of the allow synonyms values
4. One of the logical operators

A maximum of 30 sets of values can be specified. Each set must be enclosed in parentheses.

Element 1: Phrase

'phrase': Specify a phrase of one or more words, which the system compares to entries in the text search index. A maximum of 50 characters can be specified. When specifying phrases:

- an asterisk (*) can be used to mask a whole word within a phrase. For example, when searching for documents containing references to various annual reports, the following phrase can be specified:

annual * report

The search results will include documents containing such phrases as annual budget report, annual progress report, and annual sales report. The search results will also include documents containing the phrase 'annual report' without a word in between.

When using a word mask, a word must be specified before and after the asterisk. A word mask at the beginning or end of a phrase is ignored.

- an asterisk (*) can be used to mask part of a word within a phrase. The mask can be used at the beginning, middle, or end of a word. For example, when searching for documents containing references to word processing, the following phrase can be specified:

word process*

The search results will include documents containing such phrases as word processing, word processor, and word processed.

- a question mark (?) can be used to mask one or more characters in a word. For example, when searching for documents containing references to the various spellings of Johnson, the following phrase can be specified:

j?hns?n

The search results will include documents containing such phrases as Johnson, Johnsen, and Jahnsen.

Element 2: Type of Phrase

***ALL:** The phrase must be contained within one sentence, but the words do not have to be in the specified order.

***EXACT:** The phrase must be contained within one sentence and the words must be in the specified order.

Element 3: Synonyms

***NO:** No synonyms are used when searching for documents.

***YES:** Synonyms for each word in the phrase, if available, are used to compare entries in the text index.

Note:

Using synonyms may affect the performance of the request by causing more words to be searched for and by possibly causing more documents to be selected.

Element 4: Logical Operator

***OR:** If the phrase on either side of the *OR value is found, the document is selected.

***AND:** If the phrases on both sides of the *AND value are found, the document is selected.

***ANDNOT:** If the phrase following the *ANDNOT value is not found, the document is selected.

TXTLANGID

Specifies the language identifier of the document for which the user is searching. This parameter is not allowed if the QRYTXT parameter is not specified.

***JOB:** The language identifier specified for the job in which this command is entered is used.

language-identifier: Specify the language identifier.

ORDER

Specifies the order in which selected documents are placed in the created document list object and the output file (if OUTFILE is specified). For example, if *SUBJECT is specified, the selected documents are returned in order, based on the collating sequence of the document subject.

When a user specifies an order to the search request, the performance of the request may be affected. The request performs best if an order is not specified. Up to 5 document profile parameters can be specified.

Element 1: Method of Order

***NONE:** No order is applied to the selected documents.

***DOCD:** The returned documents are ordered by the document name profile parameter.

***DOCCLS:** The returned documents are ordered by the document class profile parameter.

***SUBJECT:** The returned documents are ordered by the subject profile parameter.

***EXPDATE:** The returned documents are ordered by the expiration date profile parameter.

***FILDATE:** The returned documents are ordered by the file date profile parameter.

***CRTDATE:** The returned documents are ordered by the create date profile parameter.

***ACTDATE:** The returned documents are ordered by the action due date profile parameter.

***CMPDATE:** The returned documents are ordered by the completion date profile parameter.

***CHGDATE:** The returned documents are ordered by the last document change date.

***USEDATE:** The returned documents are ordered by the last used date.

***DOCDATE:** The returned documents are ordered by the document date profile parameter.

***DOCTYPE:** The returned documents are ordered by the document type profile parameter. Valid values range from 2 through 65535.

***IDXDATE:** The returned documents are ordered by the last indexed date profile parameter. Text search services must be installed if this value is specified.

***REVDATE:** The returned documents are ordered by the last content revision date.

***KWD:** The returned documents are ordered by the keyword profile parameter.

***AUTHOR:** The returned documents are ordered by the author profile parameter.

***CPYLST:** The returned documents are ordered by the copy list profile parameter.

***OWNER:** The returned documents are ordered by the owner profile parameter.

***REF:** The returned documents are ordered by the reference profile parameter.

***STATUS:** The returned documents are ordered by the status profile parameter.

***PROJECT:** The returned documents are ordered by the project profile parameter.

***ASP:** The returned documents are ordered by the auxiliary storage pool ID (ASPID) parameter.

Element 2: Ascending or Descending Order

***ASCEND:** The returned documents are ordered in the ascending collating sequence.

***DESCEND:** The returned documents are ordered in the descending collating sequence.

CMDCHRID

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


Application Display Programming  book.

Note:

The CMDCHRID parameter applies to the following parameters and means that the data is translated to the code page and character set that is common to all of the documents in the search database.

This value translates the DOCL, QRYDFN, USRID, QRYTEXT, and TEXT parameters to values for character set and code page of '697 500'.

This value translates the USRID parameter to character set and code page of '930 500'. The SNA Distribution

Services  book contains the character set and code page table for '930 500'.

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

***DEV D:** 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 value used to create the command parameters. Valid values range from 1 through 999.

Example for QRYDOCLIB

```
QRYDOCLIB USRID(*CURRENT) OUTFILE(*NONE)
  QRYDFN(*IF ((*DOCD *EQ DOCDESC *AND)
  (*DOCCLS *BG CLASS *OR) (*FILDATE *LE '06/13/88'))))
  DOCL(MYLIST)
```

This command searches for documents that meet the following search conditions: document description equals DOCDESC and document class starts with Class; or the file date is on or before 06/13/88. The results of the search will be stored in the document list MYLIST.

Error messages for QRYDOCLIB

***ESCAPE Messages**

CPF900B

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

CPF900C

Sign on and verify of user failed.

CPF905C

Error occurred trying to find a translation table.

CPF905D

Query of document library failed.

CPF9096

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

CPF9860

Error occurred during output file processing.

QRYPRBSTS (Query Problem Status) Command Description

QRYPRBSTS Command syntax diagram

Purpose

This command allows the retrieval of problem status information from *IBMSRV (RETAIN) or from another iSeries 400 that is enlisted as a service provider.

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

Required Parameter

PRBID

Specifies the problem identifier of the problem log entry. Problems with different system origins can have the same identifier. This parameter can be used with the ORIGIN parameter to select a single problem from a particular system origin.

Optional Parameters

ORIGIN

Specifies the node of the system from which the problem log entry originated. This parameter is used with the PRBID parameter to uniquely identify the problem.

Element 1: Network Identifier

***NETATR:** The LCLNETID value specified in the system network attributes is used.

network-identifier: Specify a network identifier.

Element 2: Control Point Name

***NETATR:** The LCLNETID value specified in the system network attributes is used.

control-point-name: Specify a control point name.

SRVID

Specifies the service identifier for the problem log entry. This number is assigned when the problem is reported to IBM service support.

service-identifier: Specify the service-assigned number for the problem log entry.

RMTCPNAME

Specifies the destination of the service provider to whom the service request is sent.

Element 1: Remote Control Point Name

***IBMSRV:** The service request is sent to IBM service support.

***SELECT:** A list of service providers is shown from which the user can select the destination the service request is sent to.

remote-control-point-name: Specify the name of the control point that is the destination of the request.

RMTNETID

Specifies the remote name of the service provider's network.

***NETATR:** The service provider is in the local network.

remote-network-identifier: Specify the network name of the service provider to whom the request is sent.

AUTOPRBCRT

Specifies whether a problem should automatically be created, if a problem does not exist on the system. This will be useful if the only thing the customer has is a PMR number.

***YES:** Create a problem.

***NO:** Do not create a Problem.

Examples for QRYPRBSTS

Example 1: Querying Problem Status on Another System

```
QRYPRBSTS PRBID(1234567890) RMTCPNAME(SYSTEM99)
RMTNETID(IBMNETID) AUTOPRBCRT(*YES)
```

This command searches for the status of a specific problem on another system (SYSTEM99).

Example 2: Querying IBM Service

```
QRYPRBSTS PRBID(*PMR) RMTCPNAME(IBMSRV)
RMTNETID(*NETATR) AUTOPRBCRT(*YES)
```

This command searches the IBM Service database for the status of PMR 8X123.

Error messages for QRYPRBSTS

***ESCAPE Messages**

CPF7AA7

Problem &1 not found or in use.

CPF7AD4

Network ID &1 not in correct format.

CPF7A84

Query status request routed to different system than specified.

CPF7A88

Error indicated in reply to request.

CPF7A9A

Remote control point and network identifier not valid.

CPF7A9B

Problem &1 cannot be queried.

CPF7A97

Invalid service identifier.

CPF7A98

Service identifier not allowed.

CPF7A99

Query must be sent to *IBMSRV.

CPF7B18

Control point &1 not in correct format.

CPF8C08

Cannot specify *SELECT for the control point name.

CPF8C09

&1 not defined as a service provider.

CPF8C24

Error occurred while processing request.

QRYTIEF (Query Technical Information Exchange File) Command Description

QRYTIEF Command syntax diagram

Purpose

The Query Technical Information Exchange File (QRYTIEF) command allows the user to determine whether any files are available to be received from the remote support network. A message is returned that specifies the size (number of records) of the largest file that is to be received.

There are no parameters for this command.

Example for QRYTIEF

QRYTIEF

This command sends a message that specifies the number of files to be received from the remote support network and the size of the largest file to be received.

Error messages for QRYTIEF

None



Printed in U.S.A.